ERDC/CHL TR-18-XX

Coastal and Hydraulics Laboratory

US Army Corps of Engineers®
Engineer Research and Development Center

Adaptive Hydraulics 2D Shallow Water (AdH-SW2D) User Manual (Version 4.6)

Guidelines for Solving Two -Dimensional Shallow Water Problems with the Adaptive Hydraulics Modeling System

Compiled by The Coastal and Hydraulics Laboratory

XXX 2018

ERDC/CHL TR-18-XX
XXX 2018

Adaptive Hydraulics 2D Shallow Water (AdH-SW2D) User Manual (Version 4.6)

Guidelines for Solving Two -Dimensional Shallow Water Problems with the Adaptive Hydraulics Modeling System

Compiled by The Coastal and Hydraulics Laboratory

Coastal and Hydraulics Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Abstract: Guidelines are presented for using the US Army Corps of Engineers (USACE) Adaptive Hydraulics (AdH) modeling software to model two-dimensional shallow water problems. Constituent (non-sediment) transport is also included in this document. Sediment transport instructions are contained in a supplemental user guide.

Table of Contents

List of Figures and Tables

1 Introduction

The Adaptive Hydraulics (AdH) code is a finite element, numerical modeling package that can be used to model a wide-range of flow conditions. This includes both saturated and unsaturated 3D groundwater flow, 2D overland flow, 3D Navier-Stokes flow, 3D shallow water flow, and 2D (depth averaged) shallow water flow. However, the information contained in this manual is intended for application to the 2D shallow water flow module only.

AdH can be used in a serial or multiprocessor mode and on various operating systems after appropriate compilation. Separate executables are available for multiprocessor and serial use as well as for running on different operating systems.

The adaptive feature of AdH consists of its ability to dynamically refine and relax the mesh and temporal resolution such that both model accuracy and model performance are optimized.

The ability of AdH to allow the domain to wet and dry as flow conditions or tides change is suitable for shallow marsh environments, beach slopes, floodplains and the like.

AdH can simulate subcritical and supercritical flow conditions within the same domain. Boundary conditions can also be specified for both flow conditions.

AdH has the ability to simulate several special conditions that are pertinent to many common shallow water problems, including vessel movement, ice cover, the influence of bridge decks, the influence of culvert entrances and the presence of flow control structures such as weirs, flap and sluice gates, etc.

AdH can simulate the transport of conservative tracers, salinity, water temperature, and sediment transport that is coupled to bed and hydrodynamic changes.

AdH is developed and maintained at the Coastal and Hydraulics Laboratory (CHL) and has been used to model such varied conditions as sediment transport in sections of the Mississippi River (Heath et al., 2015; Sharp et al., 2013), tidal conditions in southern California (Tate et al., 2009), vessel traffic in the Houston Ship Channel (Tate and Ross, 2009), dam breaks (Savant et al., 2011), large scale terrestrial flooding (Tate et al. 2012) along with numerous other applications.

In General AdH-2D can be applied to problems such as the following:
1) Subcritical riverine or channel flows
2) Supercritical riverine or channel flows
3) Tidal flows
4) Dam and levee break flows
5) Flooding due to overland flow conditions: urban and others
6) Baroclinic transport (salinity, heat, as well as sediment induced)
7) Sediment transport

The code is designed to work in conjunction with the DoD Modeling Systems (XMS), developed and distributed by Aquaveo, Inc. The Surface Water Modeling System (SMS) is a modeling package for building models, running simulations, and visualizing results. The Computational Model Builder (CMB), developed by ERDC and Kitware, Inc., has also been adapted to work with AdH. However, the model setup and results can be performed through many different tools as long as the file formats are correct.

Additional information on AdH can be obtained at the AdH website, https://chl.erdc.dren.mil/chladh. The basic 2D shallow water equations and their formulation in AdH are provided in Savant et al. 2017.

1.1 Files Needed to Run AdH

Three input files are necessary to run a 2D shallow water model in AdH. The necessary files are the mesh file, the boundary condition file and the hotstart (i.e. initial conditions) file.

The **mesh** file must be constructed first and can be generated directly with GMS, SMS, CMB, or other tools that provide element and node information in the appropriate format. The mesh file defines the finite element mesh, by assigning coordinates and elevations to nodes located at the vertices of the elements, and defining a nodal connection table, indexed by the element numbers, that defines the element mesh. The node numbering convention, taken from the mesh file, is also used for defining the locations of applied boundary conditions in the boundary condition file.

Once a mesh file has been constructed, the boundary conditions for the problem and operating parameters for AdH must be specified in the **boundary condition** file. Cards are required to define the mesh-specific properties and the time-varying conditions that will drive the model.

The **hotstart** file is then generated to establish the initial conditions of the problem. All 2D models must have initial depths provided in the hotstart file. Although some areas in a 2D model can be dry initially, there must be water in the

Sign Convention

The sign convention in AdH is the standard Cartesian coordinate system and flow into the control volume is positive (section 4.8).

A note on units

AdH is designed so that the user can choose the unit system based on the values supplied. This requires that all parameters must be consistent in that they are all given in English units or SI units and not mixed.

The geometry file, boundary condition file, and hotstart file must all be given in the same unit system. There is no card that directly specifies the units being used. Rather, AdH uses the values given and calculates with them. When equations internal to AdH are unit specific, the density and gravity terms are used to decipher which system is being used.

The user can however, specify arbitrary units for gravity and other constants as well. If this is done, the user must be careful to maintain consistency between specified units and other specified input.

This manual will give unit specifications where necessary in dimensional form. **The exception to allowing the user to determine the unit system is when sediment transport is being simulated.** The sediment transport equations employed in SEDLIB are valid for SI units only. This means that all of the input files must also be given in SI units.

model in at least one element. Parameters not provided in the hotstart file are assumed to be zero unless defined specifically in the boundary condition file. All values in the hotstart file will override any parameters in the boundary condition file.

Once the required files have been created, the code can be run. All input files must have the same root name. The command to run a serial version of AdH is:

adh *filename*

AdH allows the user to provide a descriptor to the output file names. This descriptor is included with the dataset name inside the output files. This is accomplished as follows:

adh *filename descriptor*

where *filename* is the root name of the required input files. After the model is run, the results can be visualized with several tools such as SMS, GMS, Paraview, or TecPlot, as well as others.

2 Mesh file

AdH mesh files can be generated quickly and efficiently using GMS or SMS, as well as CMB. However, if these software packages are unavailable, scripts can be written to generate an AdH mesh with the necessary format. AdH meshes are unstructured in form so the element sizes can vary throughout the model domain.

In 2D, AdH uses linear triangles only. AdH can use the mesh files generated with GMS, SMS, or CMB directly, without any modifications. The mesh files should be given a **.3dm** extension by convention, regardless of the fact that they are used for a 2D application. The mesh file defines the finite element mesh, by assigning coordinates and elevations to nodes located at the vertices of the elements, and defining a nodal connection table, indexed by the element numbers, that defines the element mesh.

For full details on how to produce mesh files using GMS or SMS, refer to the appropriate reference manual and tutorials, which can be found at http://aquaveo.com. CMB references are available at http://www.computationalmodelbuilder.org/.

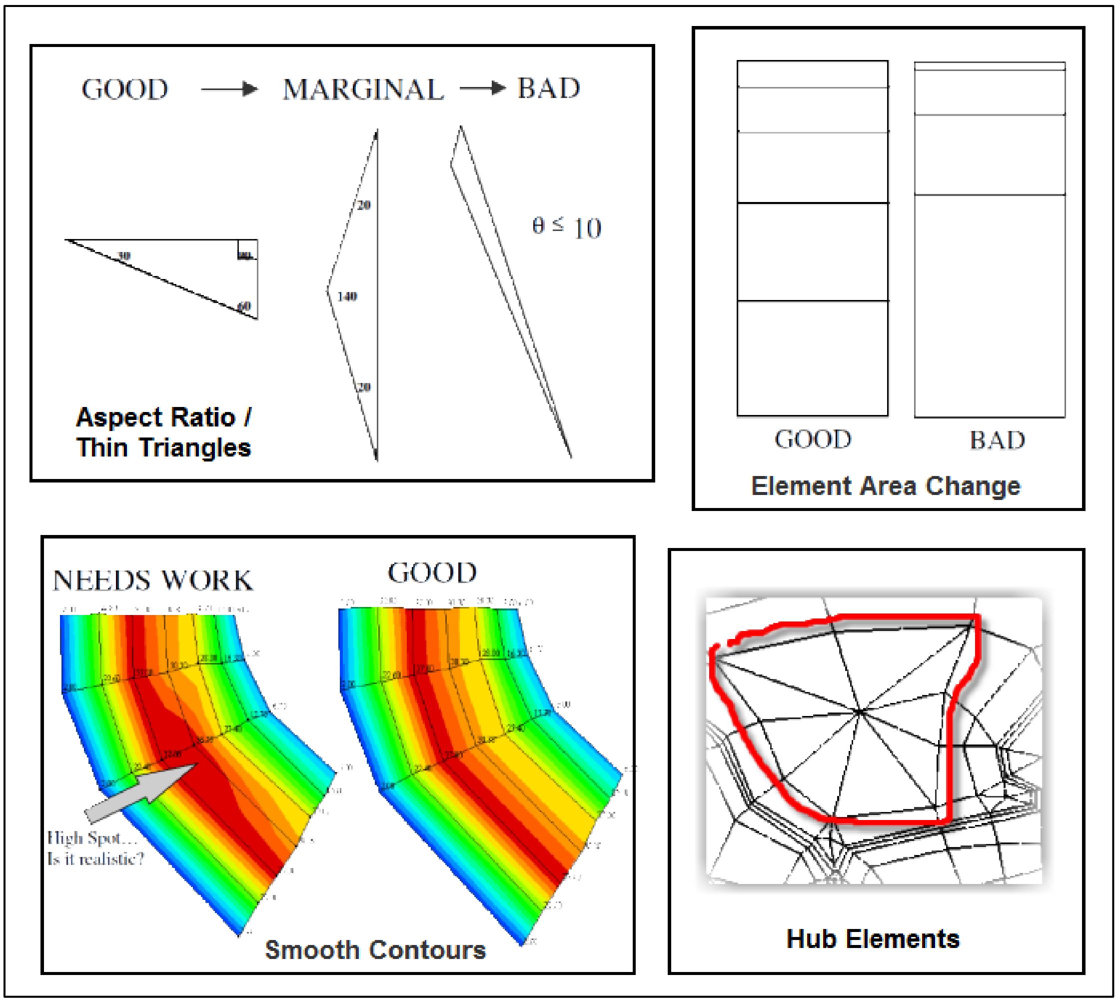
2.1 2D Mesh Files

Careful consideration of mesh element sizes, aspect ratios, and problem specifics are important when generating AdH meshes. Although AdH can adapt the mesh during the simulation period, the initial mesh must be resolved sufficiently to define the geometry of the system and important bathymetric features.

Common meshing rules apply to AdH as with any other numerical model code. Element area changes are recommended to be less than 50%. Aspect ratios of 1:3 are desired - thin triangles are typically problematic. It is also recommended that no more than 6 or 7 elements connect to a single node, often called hub elements. Good meshes include smooth bathymetric contours, smooth wet/dry boundaries, gradual element area changes, adequate resolution, mild depth changes, and a sufficiently large model domain such that boundaries are well away from the area of interest. Flow paths must have enough resolution to allow for correct flow conveyance, therefore a minimum of 3-5 elements across a channel is desired. Illustrations of some of these features of good meshing practice are provided in

Figure 1. Some references for mesh types and best meshing practice are Massey 2014 and Bakker 2012.

Figure 1. Common features of good meshing practice



Some example sections of good 2D AdH meshes are shown in Figure 2. Example sections of some poor meshing practices are shown in Figure 3. The contoured image shows a poorly defined channel connection that is created during interpolation of bathymetry to the mesh. A better defined channel or modified bathymetry can correct this issue.

Figure 2. Examples of good mesh sections

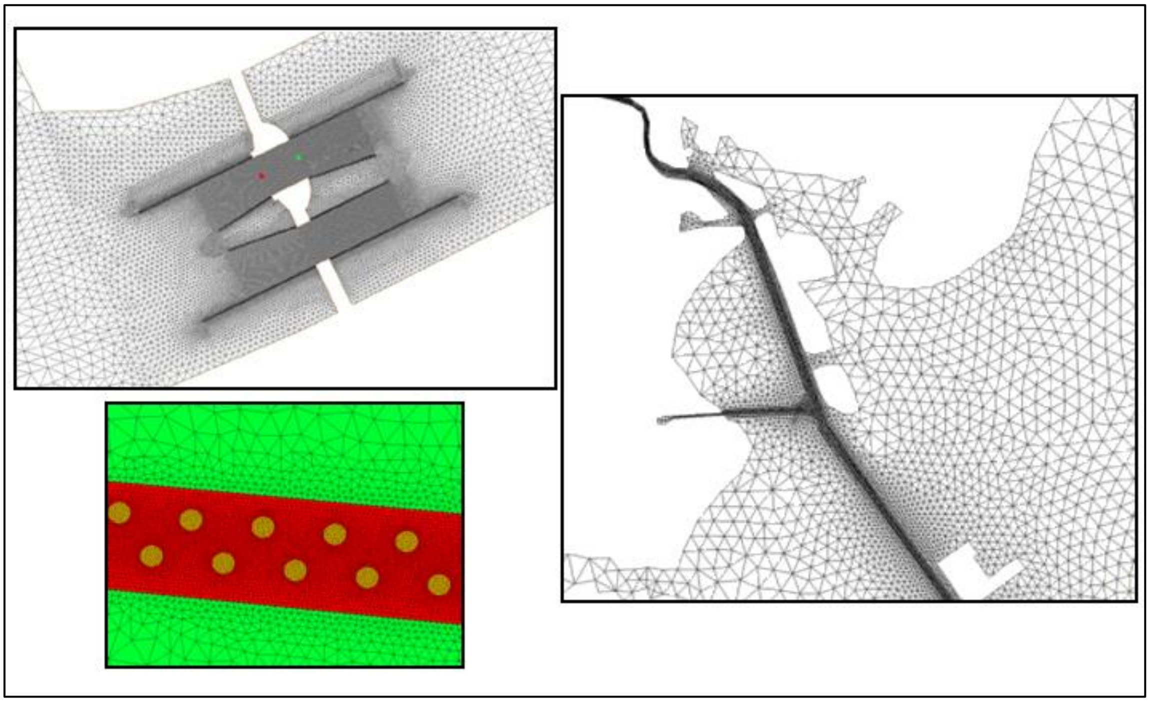
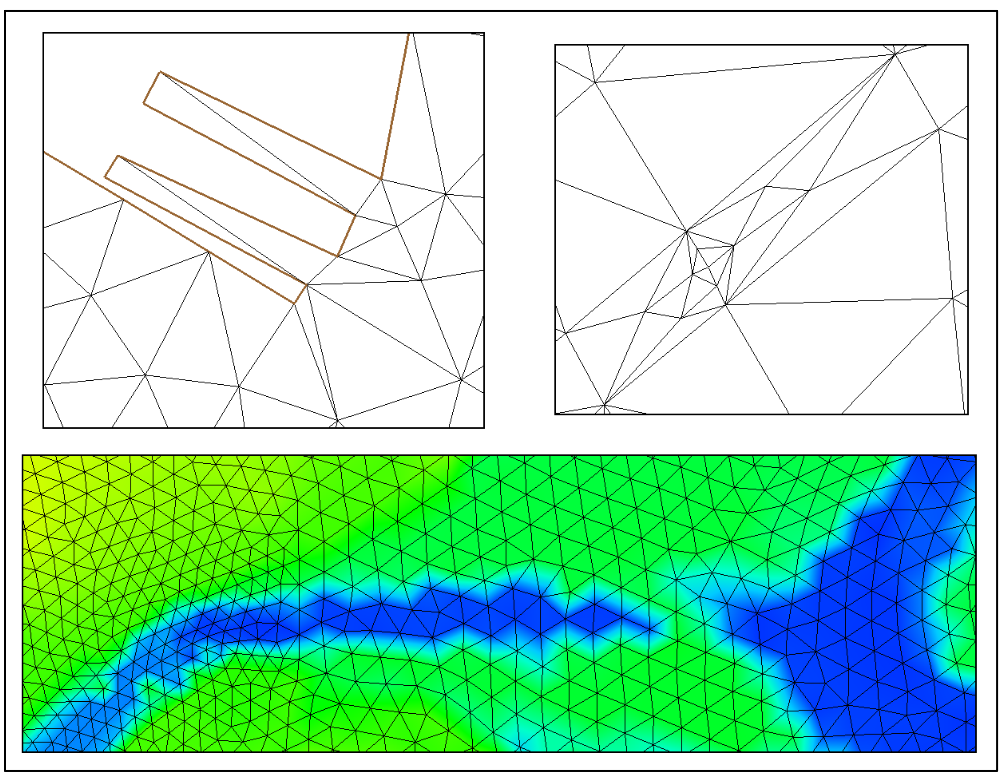


Figure 3. Examples of poor mesh definition



3 Hotstart file

The hotstart file, *filename*.hot, is used to specify initial conditions or restart conditions for the model. For hydrodynamics, a hotstart file generally consists of initial or restart conditions for velocity and depth. For transport solutions (such as sediment transport) more quantities may be specified in the hotstart file.

A hotstart file is a required input file. At a minimum, initial water depths must be provided in the hotstart file, although these depths can be zero or negative (dry). Hotstart conditions for all other variables are optional. If values for any other variables are supplied in the hotstart file, these values are used in the model simulation. Any variables that are not included in the hotstart file are initialized to zero.

A hotstart file is essentially a series of AdH result (.dat) files concatenated together to form a single file. The file format follows the standard SMS/GMS data set format. This format includes a standard set of header information followed by the values for each node in order of node number (one value per line) and ends with an **ENDDS**. No node or element numbers are given as the values are listed; there is simply the correct number of lines to match sequentially with the number of nodes. Additional datasets can then be included below the end line of the previous dataset. An example is given in Figure 4. The first two lines remain constant for all datasets. The third line can vary depending on the type of data in the dataset: **BEGSCL** for scalar data and **BEGVEC** for three column vector data. **ND** indicates the number of nodes and **NC** indicates the number of elements in the mesh. The **NAME** field provides the name for the dataset, and AdH expects uniquely defined names in order to populate each of the variables. All hotstart names are provided in Chapter 6. The optional **TIMEUNITS** flag is used by SMS when reading a dataset for visualization and can be *seconds, minutes, hours, etc.*

Figure 4. SMS/GMS data set format example

```
DATASET
OBJTYPE "mesh2d"
BEGSCL
ND   2211
NC   4000
NAME "ioh"
TIMEUNITS seconds
TS 0 0
 1
 1
 1
 .
 .
 .
 1
 1
ENDDS
```

A set of predefined dataset names is used to declare data types. An example of these is shown below.

ioh or **IOH - Initial Depth**
iv or **IV - Initial Velocity**
icon # or **ICON # - Initial Concentration**
wave_stress – Initial wave radiation stress values

Typically initial velocities of zero are acceptable, but zero depth everywhere when starting a problem will not allow water to enter the domain. There must always be some water connected to the inflow location (may be a single node) in order for the code to not assume the element is completely dry and allow the flow to enter the model domain. Negative depth values are acceptable in 2D AdH and indicate dry nodes.

Several options are available for generating initial conditions for a model simulation. The simplest initial conditions generally consist of basic starting conditions, such as level pool conditions, where initial depths are specified such that a uniform water surface elevation exists at the beginning of the model run. More complex initial conditions can also be generated, if desired. For example, if sufficient field data are available, these can be used to generate initial conditions for many problems. GMS, SMS, or CMB provide an interface for entering field data. These data can be entered into a scatter point data file and interpolated to the model mesh. Then, these data sets can be exported from GMS, SMS, or CMB and concatenated to the hotstart file in any order.

Restart conditions can be created by extracting model data at a user specified time step for each variable being simulated. These extracted data can then be concatenated together, with the appropriate headers, to generate a hotstart file.

As with initial conditions, hotstart files can also be created with GMS, SMS, or CMB for restart conditions. For a given time step, data sets for each simulated variable are exported from GMS or SMS and concatenated to the hotstart file in any order. If a dataset is not supplied for one or more of the parameters, AdH will assign default values to all the nodes for that parameter.

Another way to generate hotstart files for restart conditions is to use a utility code written by the AdH development team (adh_hot.exe). This is available for download from the AdH website.

Note that, when hotstarting, AdH reads the values in the hotstart file and assigns them at the start time specified in the boundary condition file on the **TC TO** card. The time specified in the headers of the hotstart file is ignored by AdH. For consistency, however, it is recommended that the time specified in the headers of the hotstart file — located on the **TS** line — match the start time in the boundary condition file, located on the **TC TO** card.

4 Boundary Condition File

The boundary condition file contains many pieces of information necessary to perform simulations with AdH, including:

- Model operation controls
- Applied boundary conditions
- Transport controls
- Timestep controls
- Output controls
- Adaption controls

Details of the options available for the boundary condition file are in the following sections. These options will be expanded as the AdH model continues its development and new needs arise.

4.1 Control cards

A boundary condition file, *filename*.bc, for the AdH code contains a series of one-line control cards. There is no particular order that these cards must be listed in the file, so the user can develop a method that is convenient. Cards are single line entries and cannot be wrapped across lines.

The cards fall into eleven basic categories:

- Operation parameters
- Iteration parameters
- Constituent properties
- Material properties
- Boundary strings
- Time series
- Solution controls/boundary conditions
- Friction controls
- Time controls
- Output controls
- End of file

Operation parameters control the operation of the code, the reserved memory space, type of problem being modeled, and the solver preconditioning arrangement.

Iteration parameters control the iterative methods employed by the model and the level of convergence.

Constituent properties define transport parameters for sediment and other transported quantities.

Material properties define how the model is divided up spatially in order to define properties differently in various regions.

Boundary strings are set using string array cards defining the interior and surface boundaries of the problem, including node and face boundaries.

Time series are the time varying values that will be used by the solution controls to define how parameters change over time.

Solution controls/boundary conditions specify how the conditions at the model boundaries are to be applied and vary over time.

Friction controls define the method and parameters for bed or sidewall roughness properties.

Time controls specify the timesteps used to run the model as well as the model's start and end times.

Output controls define the times at which the model results are stored for later visualization as well as the model performance output.

End of file is simply an end statement telling AdH that all input has been supplied.

4.1.1 Using control cards

Each card consists of at least one character string identifying the type of card. It may then contain further character fields and/or numeric data fields. Comments can be included in the Boundary Condition file by preceding the comment with an exclamation point (!).

4.1.2 Input file template

The minimum required information necessary for an AdH 2D shallow water boundary condition file is provided in the template below. Additional cards may be necessary depending on the specifics of the problem and the features to be included.

! Operation parameters
OP SW2 ! Physics card: 2 Dimensional Shallow Water
OP INC ! memory allocation increment
OP BLK ! blocks per processor
OP TRN ! number of transport parameters
OP PRE ! pre-conditioner selection
! Iteration parameters
IP NIT ! maximum number of non-linear iterations per group
IP MIT ! maximum number of linear iterations
IP NTL ! non-linear tolerance
and/or
IP ITL ! increment tolerance
! Material properties
MP MUC ! Manning's units constant
MP G ! acceleration due to gravity
MP MU ! kinematic molecular viscosity of water
MP RHO ! density of water
MP DTL ! wetting/drying limiting depth (defaults to 0 if not included)
MP EEV ! estimated eddy viscosity
or
MP EVS ! eddy viscosity (MP DF also required when OP TRN>0)
MP ML ! maximum number of refinement levels
MP SRT ! error tolerance for refinement (MP TRT required when OP TRN>0)
! Boundary strings (see manual for optional strings depending on specific problem)
MTS ! material strings
! Time Series
XY1 ! xy-series
! Solution Controls/Boundary Conditions (boundary condition according to specific problem)
NB or DB ! solution control options
! Friction controls (several options available)
FR ! friction options
! Time controls
TC T0 ! starting time of model simulation
TC TF ! ending time of model simulation
TF IDT ! xy-series number giving timestep size
! Output controls
OC ! timestep series defining the output frequency (requires corresponding xy-series)
or
OS ! auto-build output series defining output frequency (if used, omit OC with corresponding xy-series)
! End of file
END ! signals bc card reader that the bc file has ended

4.1.3 Control Card Categories

The hydrodynamic and transport (non-sediment) control cards are:

Table 1. Control cards

Operation Parameters	
OP SW2	2D Shallow Water REQUIRED
OP INC	Incremental Memory REQUIRED
OP TRN	Transport Quantities REQUIRED
OP BLK	Blocks per processor (used by the solver) REQUIRED
OP PRE	Preconditions (used by the solver) REQUIRED
OP BT	Enable Vessel Movement
OP BTS	Enable Vessel Entrainment
OP TEM	Enable Second Order Temporal Terms
OP TPG	Petrov-Galerkin Coefficient
OP NF2	Velocity Gradients
OP WND	Wind Stressing
OP WAV	Short Wave Stressing
OP DAM	Dam Break Stabilization
Iteration Parameters	
IP NIT	Non-Linear Iteration REQUIRED
IP NTL	Non-Linear Tolerance REQUIRED AND/OR ITL
IP ITL	Increment Tolerance REQUIRED AND/OR NTL
IP MIT	Maximum Linear Iterations REQUIRED
Constituent Properties	
CN CON	Any Constituent
CN SAL	Salinity (Baroclinic Transport)
CN TMP	Temperature (Baroclinic Transport)
CN VOR	Vorticity (Bendway Correction)
Material Properties	
MP EVS	Constant Eddy Viscosity REQUIRED OR EEV
MP EEV	Estimated Eddy Viscosity REQUIRED OR EVS
MP MU	Kinematic Molecular Viscosity REQUIRED
MP G	Gravitational Acceleration REQUIRED
MP MUC	Manning's Units Constant REQUIRED

MP RHO	Density REQUIRED
MP COR	Coriolis Latitude
MP DTL	Wetting/Drying Limit
MP DF	Turbulent Diffusion (Transport Constituent Property) REQUIRED IF MP TRT > 0 and EVS is used
MP ML	Maximum Mesh Refinement REQUIRED
MP SRT	Mesh Refinement Tolerance REQUIRED
MP TRT	Transport Refinement Tolerance (Transport Constituent Property) REQUIRED IF OP TRN > 0
MP WND STR	Wind stress specification by material
MP WND ATT	Wind attenuation specification by material
MP NVM	No vorticity by material

Boundary Strings

NDS	Node String
EGS	Edge String
MDS	Mid String
MTS	Material String REQUIRED (at least 1)

Time Series

XY1	X-Y Series REQUIRED (at least 1)
XY2	X-Y-Y Series
XYC	Wind Station Coordinates

Solution Controls

NB DIS	Natural – Total Discharge
NB OVL	Natural – Flow
NB OTW	Natural – Tailwater Elevation
NB TRN	Natural – Transport
NB SDR	Natural – Stage – Discharge Boundary
NB SPL	Natural – Spillway Boundary
NB OUT	Natural – Outflow Edge String
NB TID	Natural – Tidal Constituent Boundary
DB OVL	Dirichlet – Velocity
DB OVH	Dirichlet – Velocity and Depth
DB TRN	Dirichlet – Transport
DB LDE	Stationary Lid Elevation
DB LDH	Dirichlet – Depth of Water Under Stationary Lid
DB LID	Dirichlet – Floating Stationary Lid

DB RAD	Dirichlet – Short Wave Radiation and Dew Point Temperature Boundary
BR JAI	Dirichlet – Breach Johnson and Illes
BR SAS	Dirichlet – Breach Singh and Snorrason
BR MLM	Dirichlet – Breach Macdonald and Landgridge-Monopolis
BR FRO	Dirichlet – Breach Froelich
BR BRC	Dirichlet – Breach Bureau of Reclamation
BR VTG	Dirichlet – Breach Von Thun and Gillette
BR FER	Dirichlet – Breach Federal Energy Regulatory Commission
BR USR	Dirichlet – Breach User Defined Displacement
OB OF	Outflow Boundary
OFF	Deactivate String
WER	Number of Weirs
WRS	Weir Parameters
FLP	Number of Flap Gates
FGT	Flap Gate Parameters
SLUICE	Number of Sluice Gate
SLS	Sluice Gate Parameters

Friction Controls an FR card must be assigned for each material type, but the value can be 0 (i.e. you can assign zero friction)

FR MNG	Manning's N Roughness (Depth-corrected equation)
FR MNC	Manning's N Roughness (Traditional, or classic, equation)
FR ERH	Equivalent Roughness Height
FR SAV	Submerged Aquatic Vegetation
FR URV	Un-Submerged Rigid Vegetation
FR EDO	Equivalent Drag due to Obstructions
FR ICE	Ice Thickness
FR IRH	Ice Roughness Height
FR BRH	Bed Roughness Height
FR SDK	Submerged Dike Drag (1D friction)
FR BRD	Bridge Deck Drag (1D friction)

Time Controls

TC T0	Start Time REQUIRED
TC IDT	Time Step Series REQUIRED, OR ATF, OR STD
TC TF	Final Time REQUIRED
TC ATF	Auto Time Step Find REQUIRED, OR IDT, OR STD

TC STD	Steady State Solution
Output Controls	
OC	Output Control Series REQUIRED OR OS
OS	Auto-Build Output Series REQUIRED OR OC
FLX	Flow Output
PC ADP	Adapted Mesh Print Control
PC ELM	Numerical Fish Surrogate Output
PC LVL	Screen Output Level
PC MEO	Mass Error Output
End of File	
END	End of BC File REQUIRED (Must be last in the bc file)

4.2 Operation Parameters ⇧

Each operation parameter card consists of two character fields and may contain one numeric field. An **OP** in the first field identifies operational parameter cards. **OP** cards control the type of system being modeled. An **OP SW2** card is used to specify 2D shallow water flow modeling.

The code allocates memory as needed during the run to store the additional elements and nodes created during the refinement process. The memory is allocated in segments. The size of the segment is specified by the user on the **OP INC** card. If the specified number is too small, the program will continually seek additional memory, slowing the run time of the program. Alternately, if the number is too large, the program will require excess memory not needed to run the code.

The **OP PRE** and the **OP BLK** cards specify the preconditioner for the linear solver and the manner in which it is implemented.

```
OP SW2
OP  INC 40
OP PRE  1
OP BLK  10
```

The first card specifies the preconditioner. The integer can be 0, 1, 2, or 3 for various preconditioning schemes. The second card defines how many blocks per processor are to be used in the preconditioner. These are subdividing the problem to perform a direct solve on each block and the total group of all blocks can be used to perform a coarse grid solve. Which of these options is used is specified by the **OP PRE** choice. In this case, the 2 indicates one-level Additive Schwarz preconditioning using 10 blocks per processor. **A good starting option is PRE = 1 and BLK = 1. For version 4.6 it is recommended to always use preconditioner 1.**

After finding a flow solution, an associated transport problem can be solved. The number of transported quantities is given on an **OP TRN** card. The **OP TRN** card is a required input card. If the problem does not involve transport, zero (0) quantities are specified on the **OP TRN** card.

In addition, if transport equations are not being modeled, no transport properties or boundary conditions may be specified. An error message will be displayed if transport properties are included in the input file but no transport quantities have been specified on the **OP TRN** card. The following card specifies one transported quantity:

OP TRN 1

OP SW2

2D SHALLOW WATER PROBLEMS

Field	Type	Value	Description
1	char	OP	Card type
2	char	SW2	Specifies 2-D shallow water problem

OP TRN

TRANSPORT EQUATIONS

Field	Type	Value	Description
1	char	OP	Card type
2	char	TRN	Parameter
3	int	≥ 0	Total number of transported materials

OP INC

MEMORY INCREMENT

Field	Type	Value	Description
1	char	OP	Card type
2	char	INC	Parameter

3	int	> 0	Incremental memory allocation

OP BLK

BLOCK SPECIFICATION FOR PRE-CONDITIONER

Field	Type	Value	Description
1	char	OP	Card type
2	char	BLK	Parameter
3	int	> 0	Number of blocks per processor, used to perform pre-conditioning

OP PRE

PRE-CONDITIONER SELECTION

Field	Type	Value	Description
1	char	OP	Card type
2	char	PRE	Parameter
3	int	3 ≥ # ≥ 0	Preconditioner value 0 No pre-conditioning **1 one level Additive Schwarz pre-conditioning** 2 two level Additive Schwarz pre-conditioning 3 two level Hybrid pre-conditioning

The **OP TEM** card is an optional card that can be included to enable second order temporal terms when solving the time derivatives so that numerical dissipation is reduced. As such, higher order temporal schemes are often advantageous for problems where wave reflection is important. For example, for internal seiches, a higher order temporal scheme is beneficial, and the additional terms can often allow for larger timestep increments. Sensitivity tests of a higher scheme are encouraged.

The numerical scheme is written as the standard time-weighting scheme. The user can choose between the first or second order schemes or even a fractional amount of each by including the temporal variable. This variable, the α term below, is controlled via the **OP TEM** card and defaults to 0 if not included in the boundary condition file, indicating a first order temporal scheme. An α value of ½ is the standard Crank-Nicholson scheme. The final form of the temporal scheme is given by:

$$\frac{dh}{dt} \approx \alpha \frac{\left(\frac{3}{2}h^{n+1} - \frac{1}{2}h^{n}\right) - \left(\frac{3}{2}h^{n} - \frac{1}{2}h^{n-1}\right)}{dt} + (1-\alpha)\frac{h^{n+1} - h^{n}}{dt}$$

The following card specifies to use only the second order temporal terms since the variable is given a value of 1:

OP TEM 1

OP TEM

SECOND ORDER TEMPORAL TERM

Field	Type	Value	Description
1	char	OP	Card type
2	char	TEM	Parameter
3	real	1 ≥ # ≥ 0	Coefficient for the second order temporal scheme

The **OP DAM** card is an optional card that can be invoked when performing dam break simulations. It adds extra stability at the wetting front by reversing the sign of the petrov-galerkin temporal terms. Reversing the sign on the temporal terms in the petrov-galerkin terms is mathematically inconsistent but the stability achieved by doing so is dis-proportionately greater than the small error introduced.

OP DAM

OP DAM

DAM BREAK STABILIZATION

Field	Type	Value	Description
1	char	OP	Card type
2	char	DAM	Parameter

4.3 Iteration Parameters ⇧

There are three iteration parameter cards that must be specified by the user. Iteration parameter cards are identified by an **IP** in the first field.

An **IP NIT** card specifies the maximum number of non-linear iterations. These are the Newton iterations, used to solve non-linear equations by successive linear approximations. Typical values for the **IP NIT** card are 6 to 10.

An **IP MIT** card specifies the maximum number of linear iterations for each non-linear iteration. These linear iterations are associated with the iterative solver that is used to solve the linear system of equations when multiple processors and/or multiple blocks are invoked. Typical values for the **IP MIT** card are 50 to 200. Generally the linear iteration count will be in the low double digits for a stable model (note that when only one processor and block are used, there will be 0 linear iterations required).

An **IP NTL** and/or **IP ITL** card are required to tell AdH when convergence has been met. These cards specify the convergence tolerance for the non-linear iterations. The **IP NTL** card checks against the residual norm (the maximum difference for all nodes in successive computed solutions of the equations being solved, such as conservation of mass and momentum for hydrodynamics, and the advection-diffusion equation for transport) and the **IP ITL** card checks against the increment norm (the maximum difference for all nodes in successive computed values of the independent variables, such as u, v, and h for hydrodynamics, and c for transport). By default these tolerances are zero. The user may use either the **IP NTL** or **IP ITL** card to determine convergence; however, if both cards are used then convergence is determined when one of the tolerances is met, not both. For the linear iterations, the tolerance is set internally in AdH to 0.0001* **NTL**.

At the maximum number of iterations specified on the **IP NIT** or **IP MIT** cards, if the convergence tolerances are not satisfied, AdH will reduce the timestep size to ¼ of the previous value and continue the calculations. AdH will then double the timestep for each converged timestep until the maximum value is again reached. Therefore, setting the maximum iteration counts too high will cause the model to run slowly when it is having trouble converging. AdH will take all available nonlinear iterations before cutting the timestep size.

The example given below allows for a maximum of 5 nonlinear iterations, each with up to 100 linear iterations, to reach a convergence tolerance of 0.001. Therefore the maximum residual norm in the mesh must be less than 0.001 for convergence to be achieved.

```
IP NIT     5
IP NTL  0.001
IP MIT  100
```

IP NIT

NON-LINEAR ITERATIONS

Field	Type	Value	Description
1	char	IP	Card type
2	char	NIT	Parameter
3	int	≥ 1	Number of non-linear iterations per timestep, if at NIT the tolerance is not satisfied AdH will reduce the timestep and recalculate

IP MIT

LINEAR ITERATIONS

Field	Type	Value	Description
1	char	IP	Card type
2	char	MIT	Parameter
3	int	≥ 1	Maximum number of linear iterations per non-linear iteration by the interative solver. If the internal linear tolerance (0.0001* NTL) is not met in the maximum linear iterations, the solution stops and the timestep size is reduced.

IP NTL

NON-LINEAR TOLERANCE

Field	Type	Value	Description
1	char	IP	Card type
2	char	NTL	Parameter
3	real	≥ 0	Tolerance for maximum allowable solution residual for the non-linear iterations (perfect convergence has a residual of 0)

IP ITL

INCREMENT TOLERANCE

Field	Type	Value	Description
1	char	IP	Card type
2	char	ITL	Parameter
3	real	≥ 0	Tolerance for maximum allowable change in the velocity and depth solutions between non-linear iterations

4.4 Material Properties

Material property cards are identified by the designation **MP**. A few of the material property cards are global, such as gravity and density, and others are material specific. There must be a set of material specific cards for each material type in the model.

4.4.1 Global material parameters

The Manning units constant (**MP MUC**), gravity (**MP G**), density (**MP RHO**), and kinematic molecular viscosity (**MP MU**) are all global parameters and set over the entire model domain. These properties are fluid properties and independent of spatial variations within the model. Each card is followed by the value for the parameter. There are standard values for each of these, depending on the unit system being simulated. These values can be changed but they are typically kept constant for model of Earthbound prototypes. Typical values for the global material parameters are given in Table 2. All four of these parameters are required model inputs. A user can provide arbitrary global material properties that do not correspond to either S.I. or English units. For transport simulations involving sediment these material properties must correspond to S.I. units

Table 2. Typical global material parameter values

Parameter	SI value	English value
MUC	1.0	1.486
G	9.8 m/s^2	32.2 ft/s^2
RHO	1000 kg/m^3	1.94 $slugs/ft^3$
MU	9.8e-7 m^2/s	1.1e-5 ft^2/s

MP MUC

MANNING'S UNITS CONSTANT

Field	**Type**	**Value**	**Description**
1	char	MP	Card type
2	char	MUC	Parameter
3	real	> 0.0	Coefficient (1.486 for English units, 1.0 for SI standard)

MP G

GRAVITATIONAL ACCELERATION

Field	Type	Value	Description
1	char	MP	Card type
2	char	G	Parameter
3	real	≥ 0	Value of gravity induced acceleration (L/T^2)

MP RHO

DENSITY

Field	Type	Value	Description
1	char	MP	Card type
2	char	RHO	Parameter
3	real	≥ 0	Density (M/L^3)

MP MU

KINEMATIC MOLECULAR VISCOSITY

Field	Type	Value	Description
1	char	MP	Card type
2	char	MU	Parameter
3	real	≥ 0	Uniform background viscosity (kinematic molecular viscosity, units L^2/T)

4.4.2 Material specific parameters

The required material specific material parameters are intended for specifying properties that can change in space, such as adaption parameters and viscosity terms.

4.4.2.1 Eddy viscosity: definition

The eddy viscosity is a representation of turbulent shear stress that is analogous to the molecular viscosity associated with laminar shear stress. It is defined as the ratio of the internal shear (for 2D, the horizontal internal shear) to the spatial gradient of mean velocity.

There are two forms of eddy viscosity available in AdH – user specified (constant) and estimated (utilizing an equation to estimate the value). Both cannot be used for one material, but both can be used in a single model.

Note that it is often tempting for modelers to use the eddy viscosity as a tuning parameter to add stability, but values of eddy viscosity that are unrealistically high can negatively impact the quality of the solution by over-diffusing the velocity. Eddy viscosity is a physically meaningful quantity, and should be constrained to within physically justifiable limits.

4.4.2.2 Constant eddy viscosity

The kinematic eddy viscosity is expressed as a tensor in the following form:

$$\boldsymbol{EV} = \begin{bmatrix} EV_{xx} & EV_{xy} & EV_{xz} \\ EV_{xy} & EV_{yy} & EV_{yz} \\ EV_{xz} & EV_{yz} & EV_{zz} \end{bmatrix}$$

For 2D, the three values of the tensor are entered on the **MP EVS** card in the following order: EV_{xx}, EV_{yy}, EV_{xy}. If the eddy viscosity is independent of the direction of measurement, the formation is termed *isotropic*. In the isotropic case, $EV_{xx} = EV_{yy}$ and $EV_{xy}= 0$.

MP EVS

CONSTANT EDDY VISCOSITY (2D)

Field	Type	Value	Description
1	char	MP	Card type
2	char	EVS	Parameter
3	int	≥ 1	Material type ID number
4	real	> 0	E_{xx}
5	real	> 0	E_{yy}
6	real	> 0	E_{xy}

4.4.2.3 Estimated eddy viscosity

The estimated eddy viscosity is used as a means to calculate the eddy viscosity needed within the model as it runs, utilizing spatially and temporally varying properties of the flow. If the **MP EEV** card is supplied, several methods of estimating eddy viscosity can be invoked. For each method, the user will supply the coefficients needed to solve the underlying equation associated with each method.

The estimated eddy viscosity can be calculated using any one of four methods. These are given below.

Method 1 is an isotropic estimate of the eddy viscosity, as given in Rodi (1984). The equation is given as follows:

$$\varepsilon_{EVI} = \max\left(\varepsilon_{EVIC}, \varepsilon_{EVMIN}\right)$$
$$\varepsilon_{EVIC} = 2K\sqrt{C_d}\,hu$$
$$\varepsilon_{EVMIN} = 0.005K\sqrt{g\delta A}$$

The terms for the above equation are listed below:

K = a user-defined scaling coefficient (recommended= **0.5**)
C_d = the drag coefficient, as determined by the bed friction
A = the surface area of the element
δ = the wet-dry depth (from the DTL card)
h = the water depth
u = the depth averaged velocity

Method 2 is a means of estimating the eddy viscosity with 2 separate terms: an isotropic term that accounts for turbulent mixing (EVI), and an additional anisotropic term in the direction of flow that accounts for streamwise dispersion (spreading of the flow associated with the vertical variation of the velocity profile). The equations for each of these are given as follows (Note: the equation for turbulent mixing (EVI) is taken from Webel and Schatzmann (1984). The anisotropic term is derived from basic hydraulic principles):

$$\varepsilon_{EVI} = \max\left(\varepsilon_{EVIC}, \varepsilon_{EVMIN}\right)$$
$$\varepsilon_{EVIC} = 0.92K\sqrt{C_d}\,hu$$
$$\varepsilon_{EVMIN} = 0.005K\sqrt{g\delta A}$$

$$\varepsilon_{SDA} = 1.3\sqrt{C_d}\,hu$$

The terms for the above equations are listed below:

K = a user-defined scaling coefficient (recommended= **0.5**)
C_d = the drag coefficient, as determined by the bed friction
A = the surface area of the element
δ = the wet-dry depth (from the DTL card)
h = the water depth
u = the depth averaged velocity

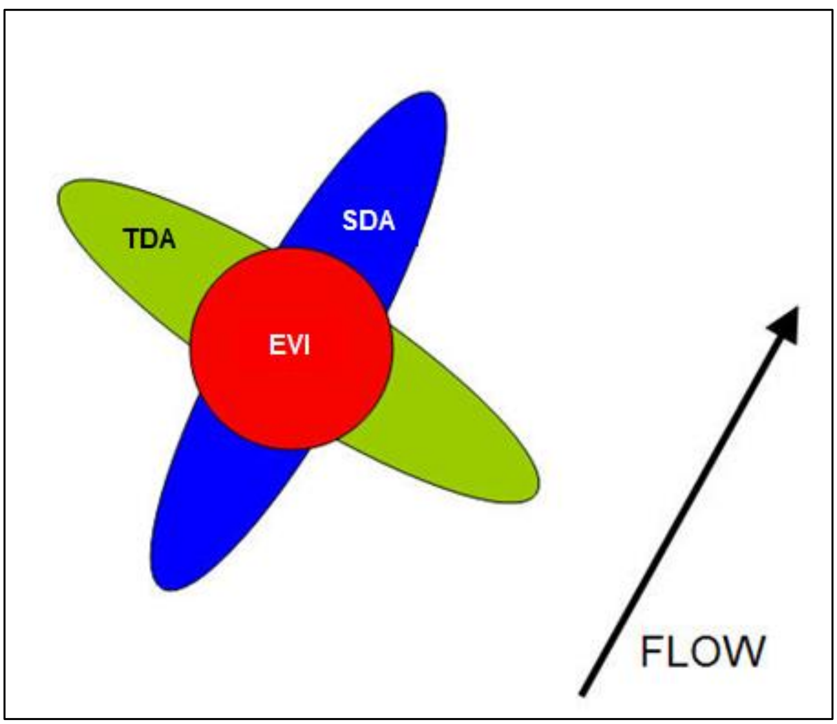
If vorticity transport (see Chapter 5, section 5.2) is active, an anisotropic dispersion term normal to the direction of flow is included in the calculations that accounts for transverse dispersion.

$$\varepsilon_{TDA} = 0.5\sqrt{C_d}\,hu_{T.MAX}$$

where $u_{T.MAX}$ = the maximum transverse velocity (taken from the vorticity calculations)

Figure 5 illustrates the various diffusion and dispersion terms associated with the application of Method 2. The EVI section illustrates the isotropic turbulent diffusion that is equal in all directions. The SDA section represents the anisotropic dispersion in the streamwise direction due to water moving faster at the surface than at the bed. The TDA section represents a transverse dispersion that occurs due to flow moving around a bend and is only included when vorticity transport is activated.

Figure 5. Diffusion and dispersion in AdH



Method 3 is the Smagorinsky (1963) formulation to compute the eddy viscosity. This option utilizes the area of the element as the length scale, A, and a user specified coefficient, K. The algorithm is given below.

$$\varepsilon_{EVI} = \max\left(\varepsilon_{EVIC}, \varepsilon_{EVMIN}\right)$$

$$\varepsilon_{EVIC} = K^2 A \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)}$$

$$\varepsilon_{EVMIN} = 0.0125 K \sqrt{g \delta A}$$

The terms for the above equations are listed below:

A = the surface area of the element
δ = the wet-dry depth (from the DTL card)
K = a user-defined scaling coefficient (recommended = **0.2**)
C_d = the drag coefficient, as determined by the bed friction
h = the water depth
u = the depth averaged velocity

Lilly (1967) analytically derived the value of C to be between 0.16 and 0.20. The user supplies this value in AdH (**the default value is 0.2**).

Method 4 is a refinement of the Smagorinsky model that uses a physically justified mixing length scale rather than the element area (Stansby, 2011). It is given as follows:

$$\varepsilon_{EVI} = \max\left(\varepsilon_{EVIC}, \varepsilon_{EVMIN}\right)$$

$$\varepsilon_{EVIC} = \sqrt{L^4\left(2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) + \frac{C_d}{2}\left(\frac{hu}{15}\right)^2}$$

$$L = 0.09\left(\min\left(6h, \sqrt{A}\right)\right)$$

$$\varepsilon_{EVMIN} = K$$

The terms for the above equations are listed below:

A = the surface area of the element

L = the mixing length
C_d = the drag coefficient, as determined by the bed friction
h = the water depth
u = the depth averaged velocity
K = a user-defined minimum eddy viscosity

Note that for Method 4, ε_{EVMIN} is directly specified by the user. This should be a small value, just large enough to ensure stability for the computations. The following equation can be used to give a starting estimate for ε_{EVMIN}.

$$\varepsilon_{EVMIN} \approx \frac{g^{1/2}\delta^{3/2}}{1000}$$

Where:

δ = the wet-dry depth (from the DTL card)
g = the gravitational acceleration

MP EEV

ESTIMATED EDDY VISCOSITY

Field	Type	Value	Description
1	char	MP	Card type
2	char	EEV	Parameter
3	int	≥ 1	Material type ID number
4	real	K	Value is dependent on the method (see above)
5	int	1, 2, 3, or 4	1 for isotropic (legacy) formulation 2 for anisotropic formulation 3 for Smagorinsky 4 for Stansby

4.4.2.4 Mesh refinement (adaption)

If desired by the user, AdH can refine and relax the resolution of the model mesh during the simulation. Mesh resolution will be added by the model according to user-defined criteria (described below) and then removed when no longer needed. This feature is useful for such things as resolving moving fronts associated with constituent transport and resolving complex flows, such as flow around a structure.

The default for refinement is no mesh refinement (0 levels of mesh refinement) with tolerance values of one. Once the adaption is turned on (level of mesh refinement > 0) adaption is controlled by the user-specified adaption parameters.

The **MP ML** card is used to specify the maximum levels of mesh refinement. The level of mesh refinement is the total number of times that an original element (within the specified material type) may be split. Refinement can be turned off in a material by specifying zero (0) as the maximum level of refinement. The maximum number of elements that can be created by adaption is 2^{ML}. For example, setting the **MP ML** value to 4 means a single element can be divided into a maximum of 16 elements.

An **MP SRT** card is required to define the tolerance of hydrodynamic adaption, for each material type. An **MP TRT** card is required to define the tolerance for consistent adaption, for each constituent and material type. If the solution error on an element exceeds the refinement error tolerance given on the **MP SRT** card (for hydrodynamics) and/or the **MP TRT** card (for transport), the element is adapted (split).

In AdH, the error is defined in terms of the root mean square of the mass conservation residual at each node in the element. This technique indicates which elements require more resolution in order to properly resolve local gradients. More details on the error indicator can be found in Tate et al (2006).

The error indicator for hydrodynamics is expressed mathematically as follows:

$$K_j = \sum_{i=1}^{i=n_j} \left(\frac{\partial h_i}{\partial t} + u_i \frac{\partial h_i}{\partial x} + h_i \frac{\partial u_i}{\partial x} + v_i \frac{\partial h_i}{\partial y} + h_i \frac{\partial v_i}{\partial y} \right)^2$$

$$E_j = A_j \sqrt{K_j}$$

Where j is the element number, i is a counter for the nodes in element j, n_j is the total number of nodes in element j, A_j is the surface area of element j, h is the water depth, u is the x-velocity, v is the y-velocity, and E_j is the hydrodynamic solution error associated with element j.

The error indicator for constituent concentration is expressed mathematically as follows:

$$K_j = \sum_{i=1}^{i=n_j} \left(h_i \frac{\partial c_i}{\partial t} + u_i h_i \frac{\partial c_i}{\partial x} + c_i h_i \frac{\partial u_i}{\partial x} + v_i h_i \frac{\partial c_i}{\partial y} + c_i h_i \frac{\partial v_i}{\partial y} \right)^2$$

$$E_j = A_j \sqrt{K_j}$$

Where j is the element number, i is a counter for the nodes in element j, n_j is the total number of nodes in element j, A_j is the surface area of element j, c is the constituent concentration, h is the water depth, u is the x-velocity, v is the y-velocity, and E_j is the transport solution error associated with element j.

The larger of the hydrodynamic error or (if transport is being solved) the transport error will determine each element's value in the *filename*_err.dat file. It is this value that is used to determine whether or not an element is refined or relaxed. The value given in the *filename*_err.dat file is normalized by the refinement tolerance (given on the **MP SRT** and **MP TRT** cards). This means that values greater than 1 can be interpreted as locations where the error exceeds the refinement tolerance.

Some users prefer to examine the error results for each transported quantity separately. To facilitate this, the hydrodynamic and transport errors are stored separately in files labeled as such: *filename*_err_hydro.dat and *filename*_err_con#.dat. The values stored in these files are not normalized by the refinement tolerance: they are the actual computed values of error. Hence they must be compared to the corresponding user-supplied values of the error tolerance on the **MP SRT** and **MP TRT** cards in order to determine where adaption is occurring.

The unrefine tolerance is set within the code as 10 percent of the refine tolerance for both flow and transport. This means that when the grid solution error improves such that the error is < 10% of the tolerance defined on the **MP SRT** and **MP TRT** cards, the elements are recombined, although never coarser than the original mesh.

During normal model operations, the solutions are always saved to the original (coarse) mesh. This means that the user will not see the adapted mesh, even though the solutions that are shown are computed on the adapted mesh. If the user desires to see the adapted mesh, he/she can invoke the **PC ADP** card (see below).

The following is an example of how to set the adaption parameters. For this case, for material type 1, a maximum of 5 levels of adaption are permitted (i.e. the individual elements can be split 32 times). For material type 1, the refinement tolerance for hydrodynamics is 100, for constituent 1 is 30, and for constituent 2 is 50.

MP ML 1 5
MP SRT 1 100
MP TRT 1 1 30
MP TRT 1 2 50

The level of refinement required is dependent on the need to have a converged solution (with respect to mesh resolution) in the area of interest for the study. That is, the selected level of refinement should yield a solution that is not significantly different from the solution resulting from the next highest level of refinement. This is directly analogous to a mesh convergence exercise, which is an important part of any traditional numerical model study. Mesh convergence means that the solution resulting from the chosen model mesh is not significantly different from the solution resulting from a model mesh with increased resolution (usually doubled resolution). That is, the solution for the chosen mesh is essentially independent of the mesh resolution.

Different material types can have different levels of refinement. Some experimentation with the error tolerance is usually necessary to gain the desired level of refinement.

If desired, the adapted meshes can be output during the simulation by including a **PC ADP** card. By including this card, the mesh and associated solution files will be saved at the time step intervals specified on the output control card. The output files will be named like so: *filename*.3dm-timestep#.0, *filename*.dep-timestep#.0, *filename*.ovl-timestep#.0 which is a geometry file for each time step, the depth solution for each time step, and the velocity solution for each time step.

MP ML

REFINEMENT LEVELS

Field	**Type**	**Value**	**Description**
1	char	MP	Card type
2	char	ML	Parameter
3	int	≥ 1	Material type ID number
4	int	≥ 0	Maximum number of refinement levels

MP SRT

FLOW REFINEMENT TOLERANCES

Field	**Type**	**Value**	**Description**
1	char	MP	Card type
2	char	SRT	Parameter
3	int	≥ 1	Material type ID number
4	real	≥ 0	Error tolerance for the refinement terms

MP TRT

TRANSPORT CONSTITUENT REFINEMENT TOLERANCE

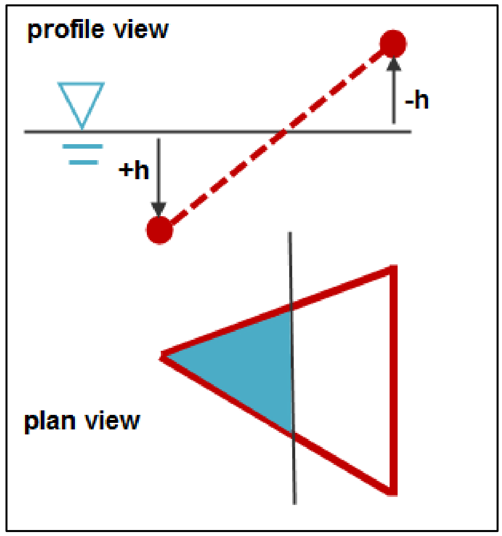
Field	Type	Value	Description
1	char	MP	Card type.
2	char	TRT	Parameter.
3	int	≥ 1	Material type ID number
4	int	≥ 1	Constituent ID number
5	real	≥ 0	Error tolerance for refinement terms

4.4.3 Optional material parameters

4.4.3.1 Wetting/drying

AdH 2D shallow water has the ability to allow areas of the mesh to become dry and then wet again as the flow varies over time. AdH is fully mass conservative for both hydrodynamics and constituent, even along the wet-dry interface. The AdH code integrates over the wet portion of partially wet elements in order to ensure a continuous wetting/drying front (Savant et al., 2017). Nodes in AdH are considered dry when the depth at the node falls below 0.0, and wet when the depth is greater than 0.0, i.e. AdH includes true wetting-drying without arbitrarily holding a certain depth of fluid. The wet-dry line is determined by a linear projection of the water depth into the dry portion of the element. The solution yields a negative depth for the dry nodes, such that the linear interpolation between the wet and dry nodes of the element results in a line of 0.0 depth that corresponds to the wet-dry line. Hence, the negative depths that appear in the depth solution arise as a result of the numerical scheme used to determine the wet-dry line.

Figure 6. Wetting/Drying illustration, h is depth



The wetting/drying depth (**MP DTL**) is used to add stability along wet/dry fronts. The wetting/drying depth specified on the **MP DTL** card (δ) does **not** represent a depth below or above which the node is dry or wet. The wetting and drying depth (δ) is an adjustment parameter applied to a stabilization term within the wet-dry logic. It is used to stabilize the solution along the wet-dry boundary. The wetting/drying depth is referred to as a “depth” simply because it has units of length, and the value specified on it should roughly correspond to the depth associated with wetting and drying regions of the domain.

Because the value of δ is proportional to the magnitude of the stabilization terms applied along the wet-dry front (and hence the diffusion of momentum along the front), very large values of δ can result in significant energy losses being incurred at wet-dry boundaries. Therefore, it is good modeling practice to run the model with the smallest value of δ that is needed for a stable, efficient solution.

The **MP DTL** value defaults to 0.0 if not included in the boundary condition file.

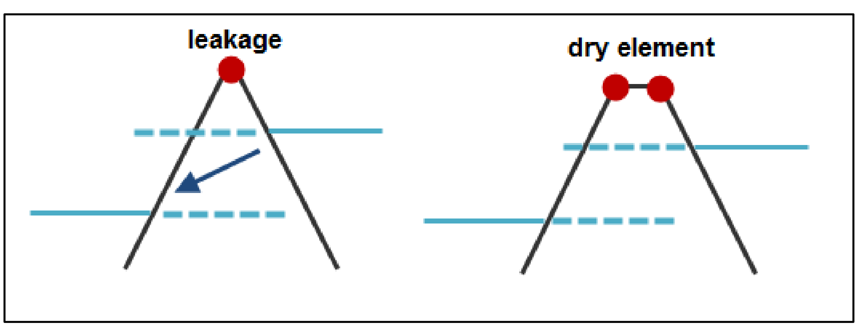
The model simulation can begin with a completely wet or partially wet domain. As time progresses it will wet and dry as necessary. The appropriate boundary locations must be partially or completely wet at the start of the simulation: flow will only pass in and out of the model through the wet portions of the boundary condition locations.

There is a specific, technical issue that is associated with the wetting/drying feature in AdH (and other models using linear elements with linear Lagrange functions), when too little resolution is used to define geometric features, especially levees. For example, consider the structure defined in Figure 7. Figure 7 shows the side-view of a structure that blocks flow under certain stage conditions (such as a levee or dike). When the structure is resolved as it is on the left of Figure 7, there are partially wet elements on both sides of the structure, but there is only one node along the crest of the structure. In this case, the AdH solution will determine a negative depth along the dry node associated with the crest, that satisfies mass conservation for both the left and right elements (and hence determines the wet dry line for both). This causes the two elements to "talk" to each other, and results in the transfer of water from one element to the other. Hence, flow can "leak" through the structure.

However, if the structure is resolved as it is on the right of Figure 7, each wet element is associated with only one dry node (completely dry elements, such as the one along the crest of the structure, are not included in the computations). Hence, the wet-dry line can be uniquely defined for each side of the structure, and there is no leakage across the structure.

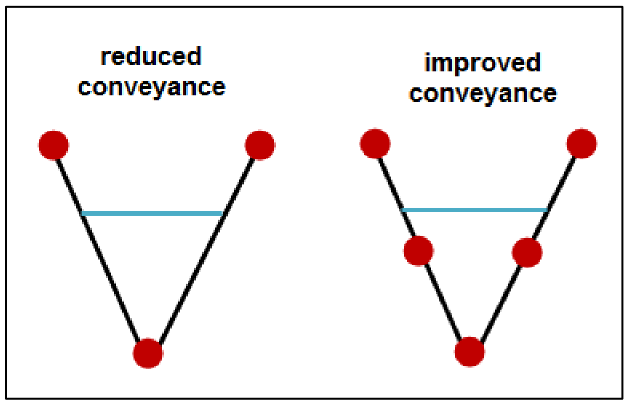
Therefore, whenever it is necessary to ensure that a structure is truly impervious to flow, the crest of the structure must be resolved with at least one element width (i.e. the crest should be two nodes wide), consisting of nodes that are all defined with the crest elevation (i.e. a flat element along the crest).

Figure 7. Low resolution leakage illustration



Another common technical issue associated with wetting/drying in AdH arises when the model defines flow pathways with too little resolution. When channels are defined with a "V" shape and are not fully wet, there are only two partially wet elements available to pass the flow. Due to the diffusion of momentum associated with partially wet elements, an apparent drag (energy loss) is associated with the flow, reducing the conveyance capacity of the channel. Therefore, a better solution can be obtained by ensuring that all significant flow paths are defined with at least 3 elements, even if the "V" shape remains (see Figure 8).

Figure 8. Channel low resolution illustration, cross section view



MP DTL

WETTING/DRYING LIMIT (2D)

Field	Type	Value	Description
1	char	MP	Card type
2	char	DTL	Parameter
3	real	≥ 0.0	wetting and drying "depth" (stabilization term), default is 0.0

4.4.3.2 Coriolis forcing

The 2-D shallow water equations also include the Coriolis force due to the earth's rotation. The **MP COR** card requires the material number and the latitude in decimal degrees for each material. This card defaults to a latitude of zero, which is associated with a Coriolis forcing of zero.

MP COR

CORIOLIS LATITUDE

Field	Type	Value	Description
1	char	MP	Card type
2	char	COR	Parameter
3	int	≥ 1	Material type ID number

4	real	-90 ≤ # ≤ 90	Latitude

4.5 Boundary Strings ⇧

For most problems, boundary conditions are needed to define the problem. Each of these boundary conditions is applied to a "string" (collection of values) consisting of element edges, nodes, or faces. Each component of a string is input on a card that specifies the string type and node or element face it contains.

There are four types of boundary strings:

- Node
- Edge (2D)
- Mid (2D)
- Material (2D)

Complete strings are input on multiple cards with one node, edge, or material per card. Cards may be input in any order and cards for different node strings may be interspersed, but this practice is not recommended. The string number will group the items together such that the same boundary conditions will be applied to all items of a specified string.

If the model has no inflow or outflow, there may be no edge or node strings to be defined. However, all models should have material strings in order to assign bed roughness to the model domain. Strings are numbered sequentially regardless of the string type. There can be gaps in the string numbering if necessary.

4.5.1 Node strings

Dirichlet (DB) data are specified on node strings. These can be made up of boundary and/or interior nodes, as the problem requires. The identifier for this card is **NDS**. On each card, the node number is followed by a string number.

4.5.2 Edge strings

Natural or flux data are specified across edge strings. Edge strings can also be used to identify a wall, i.e. solid boundary, usually for the purposes of assigning side-wall roughness. The identifier for this card is **EGS**. The card lists the identifier, two node numbers that comprise an element edge, and then the string number.

4.5.3 Mid strings

Several conditions can be defined internal to the domain across mid-strings. Mid-strings for flux computations must begin and end on a mesh boundary or in an area that is permanently dry so that accurate flow calculations can be made. Other uses for mid-strings are structure cards and some roughness options. The **MDS** card has the same format as the **EGS** card, except that it requires that elements exist on both sides of the string (rather than 1 side with the **EGS** card).

4.5.4 Material strings

These strings are used to designate a group of 2D elements for natural (NB) or flux data, such as rainfall or evaporation. They are also used to identify various spatially varying properties, such as bed roughness. The identifier for this card is **MTS**. The card lists the identifier, the material number from the mesh file and the string number. The string number does not have to be identical to the mesh material number and several mesh materials can be grouped on a single material string.

NDS

NODE STRINGS

Field	Type	Value	Description
1	char	NDS	Card type
2	int	≥ 1	ID number of a node with a Dirichlet condition
3	int	≥ 1	String ID number

EGS

EDGE STRINGS

Field	Type	Value	Description
1	char	EGS	Card type
2	int	≥ 1	ID number of the first node of an edge element
3	int	≥ 1	ID number of the second node of an edge element
4	int	≥ 1	String ID number

MDS

MID STRINGS

Field	Type	Value	Description
1	char	MDS	Card type
2	int	≥ 1	ID number of the first node of an edge element

3	int	≥ 1	ID number of the second node of an edge element
4	int	≥ 1	String ID number

MTS

MATERIAL STRINGS

Field	Type	Value	Description
1	char	MTS	Card type
2	int	≥ 1	Material type ID number
3	int	≥ 1	String ID number

4.6 Friction Controls ⇧

Friction controls are used to compute estimated values of the bed friction induced by several types of bed roughness conditions. In AdH a friction library exists that contains several different friction formulations, each of which is appropriate for characterizing different types of friction. Friction is assigned to strings. These can be material strings, edge strings, or mid strings, depending on the type of friction to be modeled. Each string is defined by one of the friction control options. If a friction parameter is not defined for a string, a value of 0.0 is applied.

Friction in AdH is always cast in terms of a drag coefficient and included as a shear stress as follows:

$$\tau_{b.x} = \frac{1}{2} C_D \rho u \sqrt{u^2 + v^2}$$

$$\tau_{b.y} = \frac{1}{2} C_D \rho v \sqrt{u^2 + v^2}$$

Each friction formulation in the AdH friction library is expressed in terms of the drag coefficient (C_D).

4.6.1 Bed Roughness

There are two equations available for computing the friction associated with bed roughness (i.e., the drag associated with open channel flow, where the drag is associated with roughness on the bed that is significantly smaller than the depth).

4.6.1.1 Theoretical log-profile roughness

The **FR ERH** card and the **FR MNG** card both invoke log-profile roughness. The log-profile roughness is the expression for C_D that results from depth-integrating the classic logarithmic velocity profile and solving for C_D. This equation, then, is the general expression for drag in turbulent, rough open channel flow. The log-profile roughness is theoretically based and therefore valid over the full range of roughness-to-depth ratios, as long as the flow is in the turbulent, rough range.

The formulation given here is derived from a modified form of the classic logarithmic velocity profile. This modified profile was physically justified by Christensen (1972). The traditional profile yields a velocity of -∞ at the bed, whereas the modified profile forces the velocity to 0.0 at the bed.

The expression is given as follows:

$$C_D = 2\left(\frac{\kappa\beta}{[(\beta+1)\{\ln(\beta+1)-1\}+1]}\right)^2$$

$$\beta = 29.7\frac{h}{k}$$

$$\kappa = 0.4$$

Where h is the water depth, k is the equivalent sand roughness height, and κ is the Von Kárman constant.

If the **FR ERH** card is used, the user supplies the value of the equivalent sand roughness height (k). If the **FR MNG** card is used, the user provides a value of Manning's n, and the code internally coverts this to a value of k using the following equation:

$$k = \left(8.25\sqrt{g}\frac{n}{K_n}\right)^6$$

$K_n = 1.0$ For Metric Units

$K_n = 1.486$ For English Units

4.6.1.2 Empirical roughness

An additional option available for computing the friction associated with bed roughness in AdH is the classic formulation of Manning's Equation. It is invoked with the **FR MNC** card.

The expression is given as follows:

$$C_D = \frac{2g}{h^{1/3}}\left(\frac{n}{K_n}\right)^2$$

$K_n = 1.0$ For Metric Units

$K_n = 1.486$ For English Units

Note that the classic form of Manning's equation was originally developed empirically, and is only approximately valid over a limited range of roughness-to-depth ratios. The log-profile roughness, however, is theoretically based, and hence is valid over the full range of roughness-to-depth ratios, as long as the flow is in the turbulent, rough range.

To specify bed roughness, the user invokes the desired equation with the appropriate card, followed by the parameters needed to define the roughness for that equation, for example, to use the **FR MNG** option for strings 1-3, add the following cards:

```
FR MNG 1 0.02
FR MNG 2 0.025
FR MNG 3 0.018
```

FR MNG

LOG PROFILE ROUGHNESS: MANNING'S N

Field	Type	Value	Description
1	char	FR	Card type
2	char	MNG	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	Manning's n

FR ERH

LOG PROFILE ROUGHNESS: EQUIVALENT ROUGHNESS HEIGHT

Field	Type	Value	Description
1	char	FR	Card type
2	char	ERH	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	Roughness height (k)

FR MNC

CLASSIC MANNING'S EQUATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	MNC	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	Manning's n

4.6.2 Submerged aquatic vegetation

The submerged aquatic vegetation method is used to compute the drag coefficient associated with the bottom shear stress resulting from a steady (or quasi-steady) current field over a bed consisting of submerged aquatic vegetation (**FR SAV**). The formulation given here is from Christensen (1985) with average vegetation characteristics taken from Jacobs and Wang (2003).

The equation is given as follows:

$$C_D = 2\left(\frac{\kappa\beta}{[(\beta+\lambda)\{\ln(\beta+\lambda)-1\}+1]}\right)^2$$

$$\beta = 29.7\frac{h}{k}$$

$$\lambda = 1 - 29.7\frac{t}{k}$$

$$t \cong \frac{2}{3}h_{sav}$$

$$\kappa = 0.4$$

Where C_D is the bed shear stress drag coefficient, h is the water depth, k is the roughness height of the SAV canopy, t is the apparent thickness of the near-zero velocity region induced by the presence of the SAV, h_{sav} is the undeflected stem height of the SAV, and κ is the Von Kárman constant.

Note that a good approximation for the roughness height of the SAV canopy is given by Jacobs and Wang (2003)

$$k \cong \frac{1}{10} h_{sav}$$

FR SAV

SUBMERGED AQUATIC VEGETATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	SAV	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	The roughness height of the SAV canopy (k)
5	real	≥ 0.0	Undeflected stem height (hs_{sav})

4.6.3 Unsubmerged rigid vegetation

The unsubmerged rigid vegetation method is used to compute a shear stress coefficient for use in computing the bottom shear stress resulting from a steady (or quasi-steady) current through rigid, unsubmerged vegetation (**FR URV**).

Some examples of this condition might include flow through mangrove stands, through phragmites in coastal wetlands, or through trees and other obstructions in coastal storm surge flooding or riverine flooding.

The formulation is taken from Walton and Christensen (1980) and it includes both the form drag induced by flow through the obstructions, and the skin drag induced by flow over the bed.

The equation is given as follows:

$$C_D = \frac{0.32\left(1\text{-}m\frac{\pi}{4}d^2\right)}{\left[\ln\left(\frac{10.94h}{k}+1\right)\right]^2} + C_{D.S}hmd$$

$$C_{D.S} = 0.4$$

Where C_D is the bed shear stress drag coefficient, h is the water depth, k is the equivalent sand roughness height, $C_{D.S}$ is the drag coefficient for the stems, d is the average stem diameter, and m is the average stem density (stems per unit area).

FR URV

UN-SUBMERGED RIGID VEGETATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	URV	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Bed Roughness Height (not including the stems) (k)
5	real	> 0.0	Average stem diameter (d)
6	real	> 0.0	Average stem density (m)

4.6.4 Evenly Distributed Obstructions

The evenly distributed obstructions method will compute a shear stress coefficient for use in computing the shear stress resulting from a steady (or quasi-steady) current through or over an evenly distributed field of flow obstructions. This card (**FR EDO**) is the most general form of form roughness drag available in AdH. It can be used to simulate flow through or over wetland vegetation, trees, buildings, or any other subgrid-scale obstructions. The obstructions are modeled as a field of evenly distributed cylinders that can be overtopped.

The formulation given here is a combination of the **FR URV** formulation taken from Walton and Christensen (1980), and the **FR SAV** formulation taken from Christensen (1985) and Jacobs and Wang (2003).

First, a drag coefficient is computed with the assumption that the obstructions are unsubmerged (**FR URV)**. Then a drag coefficient is computed with the assumption that the obstructions are submerged (**FR SAV)**. The final computed drag coefficient is selected to be the minimum of these two values

The drag coefficient associated with unsubmerged obstructions is given as follows:

$$C_{D.U} = \frac{0.32\left(1\text{-}m\frac{\pi}{4}d^2\right)}{\left[\ln\left(\frac{10.94h}{k_B}+1\right)\right]^2} + C_{D.SO}hmd$$

$$C_{D.SO} = 0.4$$

Where $C_{D.U}$ is the bed shear stress drag coefficient for unsubmerged conditions, h is the water depth, k_B is the equivalent sand roughness height of the bed, $C_{D.SO}$ is the drag coefficient for a single obstruction, δ is the average obstruction diameter, and m is the average obstruction density.

The drag coefficient associated with submerged obstructions is given as follows:

$$C_{D.S} = 2\left(\frac{\kappa\beta}{\left[(\beta+\lambda)\{\ln(\beta+\lambda)-1\}+1\right]}\right)^2$$

$$\beta = 29.7\frac{h}{k_C}$$

$$\lambda = 1 - 29.7\frac{t}{k_C}$$

$$t \cong \frac{2}{3}h_{OBS}$$

$$\kappa = 0.4$$

Where $C_{D.S}$ is the bed shear stress drag coefficient associated with submerged obstructions, h is the water depth, k_C is the equivalent sand roughness height of the obstruction canopy (when they are submerged), t is the apparent thickness of the near-zero velocity region near the bed induced by the presence of the obstructions, h_{OBS} is the average height of the obstructions, and κ is the Von Kárman constant.

The approximate value given for t as a function of h_{OBS} is taken from Jacobs and Wang (2003). Note that a good approximation for the roughness height of the SAV canopy is also given by Jacobs and Wang (2003)

$$k_C \cong \frac{1}{10} h_{sav}$$

FR EDO

UN-SUBMERGED RIGID VEGETATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	URV	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Bed Roughness Height (not including the obstructions) (k_B)
5	real	> 0.0	Canopy Roughness Height (k_C)
6	real	> 0.0	Average diameter of the obstructions (d)
7	real	> 0.0	Average height of the obstructions (h_{OBS})
8	real	> 0.0	Average density of the obstructions (number/unit area) (m)

4.6.5 Ice Friction

AdH has the ability to account for the effects that stationary ice on the water surface has on the flow below. The ice is applied as a pressure field on the water surface. The friction associated with the ice cover is modeled with a method derived from first hydraulic principles (Savant et. al., 2009). This method partitions the friction between the ice surface and the bed, allowing for sediment transport simulations under ice (if desired). The ice covered area can be defined by a material string. If the string number specified does not correspond to a material an error will result.

Once a material string is defined, it is referenced on three additional cards: **FR ICE** card to give the ice thickness and density, **FR IRH** card to give the ice roughness height, and **FR BRH** card to give the bed roughness height. These cards define the necessary parameters for the friction library to accurately account for this type of roughness element.

The equations for ice roughness are given below:

$$\tau_x = \tau_{BED.x} + \tau_{ICE.x}$$

$$\tau_y = \tau_{BED.y} + \tau_{ICE.y}$$

The shear stresses in x- and y-directions are given as follows:

$$\tau_{BED.x} = \frac{1}{2}\rho C_{D.BED}\left(\alpha_{BED} v_x\right)\sqrt{\left(\alpha_{BED} v_x\right)^2 + \left(\alpha_{BED} v_y\right)^2} + \tau_{CPE.x}$$

$$\tau_{BED.y} = \frac{1}{2}\rho C_{D.BED}\left(\alpha_{BED} v_y\right)\sqrt{\left(\alpha_{BED} v_x\right)^2 + \left(\alpha_{BED} v_y\right)^2} + \tau_{CPE.y}$$

$$\tau_{ICE.x} = \frac{1}{2}\rho C_{D.ICE}\left(\alpha_{ICE} v_x\right)\sqrt{\left(\alpha_{ICE} v_x\right)^2 + \left(\alpha_{ICE} v_y\right)^2} - \tau_{CPE.x}$$

$$\tau_{ICE.y} = \frac{1}{2}\rho C_{D.ICE}\left(\alpha_{ICE} v_y\right)\sqrt{\left(\alpha_{ICE} v_x\right)^2 + \left(\alpha_{ICE} v_y\right)^2} - \tau_{CPE.y}$$

$$v_{MAX.BED} = \frac{1}{\kappa}\ln\left(\beta_{BED}\right)u_{f.BED} \qquad v_{MAX.ICE} = \frac{1}{\kappa}\ln\left(\beta_{ICE}\right)u_{f.ICE}$$

$$u_{f.BED} = \sqrt{\frac{\tau_{BED}}{\rho}} \qquad u_{f.ICE} = \sqrt{\frac{\tau_{ICE}}{\rho}}$$

$$\beta_{BED} = 29.7\frac{z_{mv}}{k_{BED}} + 1 \qquad \beta_{ICE} = 29.7\frac{\left(h - z_{mv}\right)}{k_{ICE}} + 1$$

$$\kappa = 0.4$$

$$z_{mv} = \frac{1}{\left(A^2 + 1\right)}h$$

$$A = \frac{\ln\left(\beta_{BED}\right)}{\ln\left(\beta_{ICE}\right)}$$

$$C_{D.BED} = 2\left(\frac{\kappa\left(\beta_{BED} - 1\right)}{\left[\beta_{BED}\left\{\ln\left(\beta_{BED}\right) - 1\right\} + 1\right]}\right)^2 \qquad C_{D.ICE} = 2\left(\frac{\kappa\left(\beta_{ICE} - 1\right)}{\left[\beta_{ICE}\left\{\ln\left(\beta_{ICE}\right) - 1\right\} + 1\right]}\right)^2$$

$$\alpha_{BED} = \left(\frac{1}{\alpha_{IBR}\left(\frac{h - z_{mv}}{h}\right) + \frac{z_{mv}}{h}}\right) \qquad \alpha_{ICE} = \left(\frac{1}{\left(\frac{h - z_{mv}}{h}\right) + \left(\frac{1}{\alpha_{IBR}}\right)\frac{z_{mv}}{h}}\right)$$

$$\alpha_{IBR} = \left(\frac{C_{D.BED}}{C_{D.ICE}}\left(\frac{h - z_{mv}}{z_{mv}}\right)\right)^{1/2}$$

$$\tau_{CPE.x} \cong \rho\left(\frac{\varepsilon_{MAX.BED}+\varepsilon_{MAX.ICE}}{2}\right)\left(\frac{\left(\alpha_{BED}-\alpha_{ICE}\right)}{h\left(1-\delta_{mv}\right)}\right)v_x$$

$$\tau_{CPE.y} \cong \rho\left(\frac{\varepsilon_{MAX.BED}+\varepsilon_{MAX.ICE}}{2}\right)\left(\frac{\left(\alpha_{BED}-\alpha_{ICE}\right)}{h\left(1-\delta_{mv}\right)}\right)v_y$$

$$\varepsilon_{MAX.BED} = \frac{1}{4}\kappa u_{f.BED} z_{mv} \qquad \varepsilon_{MAX.ICE} = \frac{1}{4}\kappa u_{f.ICE}\left(h-z_{mv}\right) \tag{27,28}$$

$$\delta_{mv} \cong 0.368 \tag{29}$$

LIST OF SYMBOLS FOR ICE FRICTION:

$C_{D.BED}$	=	the bed shear stress drag coefficient
$C_{D.BED.ONLY}$	=	the bed shear stress drag coefficient, assuming $z_{mv} = h/2$
$C_{D.ICE}$	=	the ice shear stress drag coefficient
g	=	the gravitational acceleration
h	=	the water depth
k_{BED}	=	the equivalent bed roughness height
k_{ICE}	=	the equivalent ice roughness height
v	=	the depth-averaged velocity magnitude
v_x	=	the depth-averaged velocity in the x-direction
v_y	=	the depth-averaged velocity in the y-direction
$v_{MAX.BED}$	=	the maximum velocity for the bed profile
$v_{MAX.ICE}$	=	the maximum velocity for the ice profile
$u_{f.BED}$	=	the friction velocity for the bed profile
$u_{f.ICE}$	=	the friction velocity for the ice profile
z_{mv}	=	the distance above the bed at which the maximum velocity is located (i.e. the location of the transition from the bed induced velocity profile to the ice induced velocity profile)
α_{BED}	=	the mean velocity correction factor for the bed shear stress
α_{ICE}	=	the mean velocity correction factor for the ice shear stress
α_{IBR}	=	the ratio of the mean velocity for the ice velocity profile to the mean velocity for the bed velocity profile
$\varepsilon_{MAX.BED}$	=	the maximum eddy viscosity for the bed profile
$\varepsilon_{MAX.ICE}$	=	the maximum eddy viscosity for the ice profile
δ_{mv}	=	the normalized fraction of the distance to the centroid of the velocity profile
κ	=	the Von Kárman constant
ρ	=	the density of water
ρ_{ice}	=	the density of ice
S_{EGL}	=	the slope of the energy grade line
τ_{BED}	=	the boundary shear at the flow-bed interface
τ_{ICE}	=	the boundary shear at the flow-ice interface

τ_{CPE}	=	the approximate cross-profile exchange of shear stress
τ	=	the total shear stress

FR ICE

ICE THICKNESS

Field	Type	Value	Description
1	char	FR	Card type
2	char	ICE	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Ice thickness
5	real	> 0.0	Ice density
6	int	= 0	0 – stationary
		= 1	1 – moving ice (not implemented yet)

FR IRH

ICE ROUGHNESS

Field	Type	Value	Description
1	char	FR	Card type
2	char	IRH	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Ice roughness height (k_{ICE})

FR BRH

ICE BED ROUGHNESS HEIGHT

Field	Type	Value	Description
1	char	FR	Card type
2	char	BRH	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Bed roughness height (k_{BED})

4.6.6 Sidewall friction

At times, models require the inclusion of losses generated by a sidewall. Sidewall losses can be important in narrow channels with steep sidewalls, such as lock chambers. Sidewall friction can also be important for resolving flow separation around an emergent object, such as a bridge pier. The sidewall friction is applied in the same manner as the bed friction options but is assigned to an edge string (**EGS**) rather than a material string.

4.6.7 1D internal friction (local losses)

When flow passes over a submerged weir, or under a bridge deck under pressure, there is a significant loss of energy associated with the expansion of the flow downstream of the obstruction. This expansion is associated with vertical flow separation, and hence is not directly simulated in a hydrostatic model.

To address this problem, AdH includes the ability to assign discrete energy losses along a 1D internal string (an **MDS** string). The loss associated with the expansion of flow downstream of a submerged dike (or weir) can be simulated by invoking a **FR SDK** card, and the loss associated with a bridge deck (either under pressure, or overtopped) can be simulated by invoking an **FR BRD** card.

For example, if the energy loss over a submerged dike is being simulated, an **MDS** string is created along the crest of the dike, and a **FR SDK** card is associated with that **MDS** string. Then, the drag is computed that is associated with the velocity vectors projected normal to the **MDS** string, and that drag is included in the model equations as a discrete loss of energy.

Note that the energy losses associated with dikes and or bridge decks can be approximated in this way without having to resolve the structures only if detailed calculations of flow or sediment transport associated with specific dikes or bridge decks are not required (for example, if they are located at some distance from the area of interest for the study). One can simply define an MDS string along nodes that define the location of the structure (the crest of a dike, or the length of a bridge), and use the 1D friction cards to extract the energy loss associated with the object from the flow. However, if detailed calculations are necessary, the structures mush be resolved geometrically.

The drag associated with the **FR SDK** card is computed as follows (Brown, 2017)

$$C_D = 2\left(\frac{a}{a+h}\right)$$

Where h is the water depth and a is the height of the dike above the bed.

The drag associated with the **FR BRD** card is computed as follows (Chu et. al. 2012)

$$h_* = \frac{\text{h-h}_\text{B}}{\text{t}_\text{B}}$$

If $h_* < 0$

$$C_D = 0$$

If $0 < h_* < 1$

$$C_D = 1.58 h_*^2 \left(1 + 128.2 \left(\frac{t_B}{t_B + h_B} \right)^4 \right)$$

if $h_* > 1$

$$C_D = 1.58 \left(1 + 128.2 \left(\frac{t_B}{h} \right)^4 \right)$$

Where h is the water depth, h_B is the height of the bottom of the bridge deck, and t_B is the thickness of the bridge deck.

FR SDK

1D INTERNAL FRICITON: SUBMERGED DIKE

Field	Type	Value	Description
1	char	FR	Card type
2	char	ICE	Parameter
3	int	> 0	Midside string ID number
4	real	> 0.0	Height of the dike (above the bed) (a)

FR BRD

1D INTERNAL FRICITON: BRIDGE DECK

Field	Type	Value	Description
1	char	FR	Card type
2	char	ICE	Parameter
3	int	> 0	Midside string ID number
4	real	> 0.0	Elevation of the bridge deck (h_B is calculated using the difference of this value and the local bed elevation)
5	real	> 0.0	Thickness of the bridge deck (t_B)

4.7 Time Series

Boundary condition data and some model control data are specified as time series. The time series are just as they sound — time values and the corresponding data values. The series may be used to define how the flow changes with time, the change in the boundary water surface elevation over time, the change in the timestep as the model runs, and even the time at which data is output for review.

All time series cards, regardless of type, follow a similar syntax. They begin with **XY1** (wind is **XY2**), followed by the series number, the number of points in the time series, the unit flag for the time values in the series, and the unit flag for output (only used if the time series defines model output). The unit specifications are as follows: 0 = seconds, 1 = minutes, 2 = hours, 3 = days, and 4 = weeks. These unit specifications only refer to the time column. They do not impact the values in the additional data columns. Below this **XY** line are the time and data points making up the series.

As the model steps through time, the values on the time series will be applied to the simulation at the elapsed time by linear interpolation of the series. Therefore, the time increment on the time series does not have to match the timestep being taken by the model. There are several different time series options.

All time series, regardless of the type, should be in time order: i.e. starting with the earliest time value and proceeding to the latest.

Also note, the initial time of each series must be less than or equal to the initial time of the simulation, and final time of each series must be equal to or greater than the final time of the simulation. This is so that the series are defined for the entire range of simulated time.

Some samples of XY series are given below.

```
XY1   1   2   0   0
0.0          1.0
2000.0       1.0

XY1   2   4   0   0
100.       0.0
200.       0.0
1600.      0.0
```

```
2000.    0.0

XY1  3  3  0  0
0.0        20.0
200.0      40.0
2000.0     100.0
```

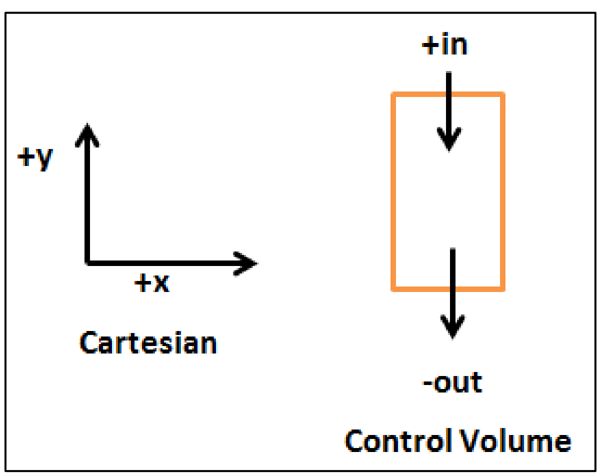
Details of each type of time series card will be provided in the appropriate section for its specific use. Boundary condition time series are in section 4.8.1; timestep time series are in section 4.9.2; and output time series are in section 4.10.1.

4.8 Solution Controls ⇧

Solution development is controlled through the specification of the initial and boundary conditions as well as the timestep parameters. AdH includes multiple solution control options, the choice of which is dependent on the problem you are solving.

For velocity components, all directions (i.e. signs) are defined by the standard Cartesian coordinate system. For flow conditions applied to an edge, material, or face, flow into the domain (i.e. control volume) is positive and flow out is negative.

Figure 9. Sign convention



Dirichlet boundary conditions are specified on a **DB** card, and Natural boundary conditions on a **NB** card. The following sections discuss how to apply the various boundary configurations.

4.8.1 Boundary condition series

Models can be set up such that no flow ever enters or leaves the domain. In that case, no boundary flow conditions are needed. However, most models will have boundary conditions that control the exchange of flow and constituent at the boundaries.

A basic boundary condition series is given as **XY1** (XY2 for wind). These series will be used for solution controls to be specified on additional cards in the boundary condition file. These series will be referenced in the solution controls section of the boundary condition file with units dependent on the type of solution control they are defining, i.e. velocity, depth, concentration.

XY1

X-Y SERIES

Field	Type	Value	Description
1	char	SERIES	Card type
2	char	BC	Series type
3	int	> 0	ID number of the series
4	int	> 0	Number of points in the series
5	int	> 0	Input units. (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)
5	int	> 0	Output units (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)

4.8.2 Flow boundary

4.8.2.1 Subcritical inflow

Several flow boundary conditions are available for subcritical flow conditions. Total discharge and unit discharge options are available for supplying or removing flow from the model domain.

A commonly used method of applying subcritical inflow is to use a discharge boundary condition (**NB DIS**). The discharge boundary condition is specified as total inflow to the model. This inflow is applied at the inflow edge string, with flow distributed according to the estimated relative conveyance of each section of the string. Hence, the model seeks to distribute the inflow as naturally as possible. *If an **NB DIS** boundary is defined, friction associated with the material on which the NB DIS boundary is applied cannot be zero*.

NB DIS boundaries are applied by first defining an **EGS** string along the inflow boundary. Then, the NB DIS string is invoked and the inflow is applied as a Natural boundary condition. The **NB DIS** string is used to associate the **EGS** string with the **XY** series that defines the inflow time series. For the following example, the **NB DIS** (discharge) boundary condition is followed by the **EGS** string number (2) and then the **XY** series number (4) defining the total inflow values over time.

```
XY1   4   2   2   0
0.0         1.0
2000.0      1.0

EGS  10 11  2
EGS  11 12  2
EGS  12 13  2

NB DIS  2  4
```

EGS string 2 is defined as including edges consisting of nodes 10, 11, 12, and 13. The discharge boundary condition is applied on this string of edges with values defined on **XY** time series number 4, which defines a constant flow rate of 1.0 L^3/T from 0.0 to 2000 minutes.

Another option for defining inflow as a natural boundary condition is to specify the flow per unit width (rather than the total inflow) using the **NB OVL** card. This option is very similar to the **NB DIS** option, except that the flow is specified as flow per unit width (rather than total discharge) and the flow is evenly distributed along the boundary rather than distributed by relative conveyance. This method of applying inflow is preferable to the **NB DIS** card when the Froude number of the inflow approaches critical flow conditions, due to the fact that the logic that distributes the flow along the boundary associated with the **NB DIS** card is unstable at high Froude numbers.

The following example shows how to invoke the **NB OVL** card on **EGS** string number 2 and defined by the values on **XY** time series number 4.

```
XY1   4   2   2   0
0.0         1.0
2000.0      1.0

EGS  10 11  2
EGS  11 12  2
EGS  12 13  2

NB OVL  2  4
```

In this case, the values on **XY** series 4 are flow per unit width, or 1.0 L^2/T over the simulation. The units on the **XY** series values are dependent on the boundary condition for which they are assigned.

It is also possible to apply velocity directly as a boundary condition, by using a Dirichlet (strongly enforced) boundary condition. Dirichlet boundaries are applied at individual nodes, and hence a node string is needed to define the boundary. In the example below, the **DB OVL** (velocity component) boundary condition is followed by the **NDS** string number and then the **XY** series number for the *x*-component and the **XY** series number for the *y*-component of velocity.

```
NDS  101  3
NDS  102  3
NDS  103  3

DB OVL  3  1  2
```

NDS string 3 is defined as including nodes 101, 102, and 103. Dirichlet boundary conditions are applied for the velocity components on string number 3. The *x*-component is defined by the **XY** time series number 1 and the *y*-component is defined by **XY** time series number 2. Recall that the direction of the velocity components follows the Cartesian coordinate system.

OB OF boundaries can be specified on edge strings for boundary locations where exact flow conditions are unknown. The **OB OF** boundary will allow the flow that touches this boundary to flow out without any modification to flow properties. This boundary condition option is necessary for a supercritical inflow (see section 4.8.2.2) since there is a discontinuity between the two locations.

NB DIS

NATURAL BOUNDARY CONDITION - TOTAL DISCHARGE

Field	Type	Value	Description
1	char	NB	Card type
2	char	DIS	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Series ID number containing the total discharge across the string; positive in

NB OVL

NATURAL BOUNDARY CONDITION - FLOW

Field	Type	Value	Description
1	char	NB	Card type
2	char	OVL	Parameter
3	int	≥ 1	String ID number (2D edge or 2D material)
4	int	≥ 1	Series ID number containing the flow data; for material strings the series values represent the flow per unit area (L/T - positive in); for edge strings the series values represent the flow per unit width (L^2/T - positive in)

DB OVL

DIRICHLET – VELOCITY (2D)

Field	Type	Value	Description
1	char	DB	Card type
2	char	OVL	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number for x-velocity component
5	int	≥ 1	Series ID number for y-velocity component

OB OF

NATURAL OUTFLOW

Field	Type	Value	Description
1	char	OB	Card type
2	char	OF	Parameter
3	int	≥ 1	String ID number (edge)

4.8.2.2 Supercritical inflow

Supercritical inflow, where Froude numbers are greater than 1.0, requires that both components of velocity and depth be defined at the inflow location. The syntax is similar to the **DB OVL** card except that the depth is also defined for the **DB OVH** card. Below is an example.

```
XY1   2   2   2   0
0.0         1.0
2000.0    1.0

XY1   3   2   2   0
0.0         -1.3
2000.0    -1.3

XY1   5   2   2   0
0.0         0.23
2000.0    0.23

NDS  221  4
NDS  232  4
NDS  223  4
NDS  126  4

DB OVH    4  1  2  5
```

Here the **NDS** node string is number 4. The Dirichlet boundary condition (**DB OVH**) is for supercritical flow. The x-component of velocity is given by **XY** series number 1 (with a constant 1.0 L/T x-velocity), the y-component by **XY** series number 2 (with a constant -1.3 L/T y-velocity), and the depth by **XY** series number 5 (with a 0.0.23 L depth). For SI units, these values provide a Froude number 0f 1.1. For a fully supercritical domain, a supercritical inflow condition at the upstream boundary will be paired with a natural outflow (**OB OF**) condition at the downstream boundary.

DB OVH

DIRICHLET - VELOCITY AND DEPTH

Field	Type	Value	Description
1	char	DB	Card type
2	char	OVH	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number for x-velocity component
5	int	≥ 1	Series ID number for y-velocity component
6	int	≥ 1	Series ID number for the depth

4.8.3 Water surface elevation boundary

In most riverine applications, the upstream boundary is specified with a flow condition and the downstream boundary is given as a tailwater elevation boundary. Tidal applications also require an elevation boundary condition that defines the change in the tide over time.

4.8.3.1 Natural water surface elevation boundary

An **EGS** string is first defined and then a **NB OTW** card is specified to link a particular time series of water surface elevation to this string. See the following example.

```
EGS  623  624  5
EGS  624  627  5
EGS  627  629  5

NB OTW  5    6
```

This group of element edges composes **EGS** string number 5. The tailwater elevations for this string are found on **XY** time series number 6.

NB OTW

NATURAL BOUNDARY CONDITION - WATER SURFACE ELEVATION

Field	Type	Value	Description
1	char	NB	Card type
2	char	OTW	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Series ID number that contains the time series of the water surface elevation

4.8.4 Atmospheric boundary (Rain or Evaporation)

The **NB OVL** card can also be used to define flows into and out of the domain from above and below (such as rainfall and evaporation) by simply invoking a material string instead of an edge string. For this case, the applied flow is the flow per unit surface area, rather than the flow per unit width.

```
MTS 1 6
MTS 2 6

FR MNG 6 0.03

NB OVL 6 7
```

This example says that material types 1 and 2 (which represent many elements in the mesh file) are grouped together and represented by material string number 6. String 6 will have a Manning's bed friction coefficient of 0.03. The inflow/outflow rate per unit area is specified on **XY** time series number 7. The values on the **XY** time series will be the total inflow/outflow divided by the surface area of the elements in the material string.

As with inflow directions on the boundary, positive values are into the control volume (rain) and negative values are out (evaporation).

NB OVL

ATMOSPHERIC SOURCE

Field	Type	Value	Description
1	char	NB	Card type
2	char	SOURCE	Parameter
3	int	≥ 1	String ID number (material or face)
4	int	≥ 1	Series ID number that contains the time series of the source values (unit flow – L/T)

4.8.5 Stage Discharge

The **NB SDR** card specifies a stage-discharge boundary dependent on the user specified power relationship in the form:

$$Q = E\left(A + C(\eta - B)^{D}\right)$$

Where Q is the discharge, η is the water surface elevation, and A through E are user specified constants.

This boundary condition option requires an edge string and relates the total flow to be applied along this edge to a stage elevation.

NB SDR

STAGE DISCHARGE BOUNDARY

Field	Type	Value	Description
1	char	NB	Card type
2	char	SDR	Parameter
3	int	≥ 1	String ID number
4	real	≥ 0	Coefficient A
5	real	≥ 0	Coefficient B
6	real	≥ 0	Coefficient C
7	real	≥ 0	Coefficient D
8	real	≥ 0	Coefficient E

4.8.6 Spillway boundary

The **NB SPL** card specifies a stage-discharge boundary dependent on the user specified percentage of flow leaving the domain, as with a spillway.

NB SPL

NATURAL BOUNDARY CONDITION - SPILLWAY

Field	Type	Value	Description
1	char	NB	Card type
2	char	SPL	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Series ID number that contains the time series of the percent (%) flow out.

4.8.7 Diversion boundary

The **NB OUT** card specifies flow withdrawal at one defined string and the same flow re-entering the domain at a second defined string. It is useful for modeling controlled diversions of flow, such as power plants and water (and sediment) diversions. As with inflow directions on the boundary, positive values are into the control volume and negative values are out. Therefore, the values on the outflow time series should be negative.

NB OUT

FLOW OUTPUT FROM INSIDE THE GRID

Field	Type	Value	Description
1	char	NB	Card type
2	char	OUT	Card type
3	int	≥ 1	Outflow edge string
4	int	≥ 1	Inflow edge string
5	int	≥ 1	Series ID number of the outflow

4.8.8 Off boundary

All strings are included in the calculations by default, so every defined string is expected to have a friction parameter or solution control associated with it. Strings can be removed from the computations by using the **OFF** card followed by the string number. When applied to material strings, this allows the modeler to add or remove sections of the domain without having to generate a new mesh or make significant changes to the boundary condition file.

OFF

DEACTIVATE STRING

Field	Type	Value	Description
1	char	OFF	Card type
2	int	> 0	String ID number

4.9 Time controls ⇧

4.9.1 Run time controls

The beginning, end, and time increment of the solution are determined by a group of cards with the time control specifier **TC**. The start time is specified on a **TC T0** (“T” zero) card. The final time, at which the run will terminate, is specified on a **TC TF** card. The final time does not have to equal the largest value in the time series but the time series must extend equal to or beyond the final run time. The units for the start and end time controls can be individually set by providing a flag after the time values (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks). The unit designation is optional: the default is seconds. The below example starts the model at zero seconds and ends the model at 2000 hours.

```
TC T0   0.0
TC TF   2000.0  2
```

TC T0

START TIME

Field	Type	Value	Description
1	char	TC	Card type
2	char	T0	Parameter
3	real	> 0	Start time of the model
4	int	#	Units (optional; 0 = seconds, 1 = minutes, 2 = hours, 3 = days, 4 = weeks)

TC TF

FINAL TIME

Field	Type	Value	Description
1	char	TC	Card type
2	char	TF	Parameter
3	real	> 0	End time of the model
4	int	#	Units (optional; 0 = seconds, 1 = minutes, 2 = hours, 3 = days, 4 = weeks)

4.9.2 Time step size

The maximum timestep size must be specified on a time series card and can be defined such that it can vary during the simulation period. An **XY** time series card is used to define the timestep maximum during the simulation. This time series is then referenced on a **TC IDT** time control card. The timestep size is always given in seconds regardless of the units of the time variability of the first column of the series.

Timesteps will be reduced if the model fails to converge for the current timestep (i.e. residuals remain greater than the convergence tolerance after the maximum number of iterations is reached). The timestep is dropped to ¼ of the previous timestep size with each failure. When a reduced timestep size does converge, however, the timestep size will double until it reaches the maximum timestep value specified in the time series referenced by the **TC IDT** card.

TC IDT

TIME STEP SIZE

Field	Type	Value	Description
1	char	TC	Card type
2	char	IDT	Parameter
3	int	> 0	Series ID number containing the length of timestep (Δt).

AdH is an adaptive code and has the ability to internally refine or relax the temporal resolution depending upon how the simulation is progressing and what the hydrodynamic conditions are. Adaptive time stepping in AdH provides two distinct options. These are the Steady State (TC STD) and the Auto-Time Step Finder (TC ATF) options.

The TC STD time stepping option is suitable to efficiently arrive at the time independent solution to steady state problems, and the TC ATF time stepping option is suitable to efficiently step through time for a rapidly varying hydrodynamic problem such as dam break flow.

When the TC STD option is specified the code steps through time without consideration to the non-linear or the linear tolerances, however the code will attempt to take the number of iterations provided on the IP NIT and IP MIT cards.

Technical details on the ATF option are provided in Savant et al. (2010).

TC STD

STEADY STATE

Field	Type	Value	Description
1	char	TC	Card type
2	char	STD	Parameter
3	int	> 0	Minimum time step size.
4	int	> 0	Maximum time step size

TC ATF

AUTOMATIC TIMESTEP FIND

Field	Type	Value	Description
1	char	TC	Card type
2	char	ATF	Parameter
3	int	> 0	Minimum time step size.
4	int	> 0	Maximum time step size series

4.10 Output control

4.10.1 Solution output times

The card that defines the times desired for output to be written is the **OC** card. The **OC** card references an **XY** time series that defines the time values at which solution data are stored (the value for every time in the series will be zero). Output is written to any of several data set files (*.dat) files.

Since AdH can vary the timestep size throughout the simulation, the times requested for output may not match those for which a solution is obtained. Therefore, the solutions are written to the *.dat files at the solution time step that corresponds to the closest elapsed time that is equal to or greater than each time requested on the output control time series. This is why the exact time requested may not appear in the solution files.

If using the auto-build feature for the output time series (see below), the **OC** card is not required. See the section on solution output files for more information on the data provided by AdH.

OC

OUTPUT

Field	Type	Value	Description
1	char	OC	Parameter
2	int	> 0	Series ID number that contains the time steps to be output

4.10.2 Auto-build output series

The auto-build feature enables an output time series to be automatically created during runtime, given certain parameters. This option eliminates the need to manually type out each timestep desired to be in the solution files. In order to activate this feature, simply use the **OS** card followed by the series number, number of time segments used to build the output series, and the time output units (i.e. the time units for the values in the saved data files). A time segment is composed of four parameters: start time, end time, progression interval, and unit flag for the times on this line. The unit specifications are as follows: 0 = seconds, 1 = minutes, 2 = hours, 3 = days, and 4 = weeks. An example is shown below.

```
OS  2     4    0
70  77.90  5.0  0
```

1.5 6.0 2.0 1
7.0 9.0 3.0 1
0.5 13.0 6.0 2

The auto-build time series above is equivalent to the times, given in seconds, below.

70.0
75.0
77.0
90.0
210.0
330.0
360.0
420.0
540.0
1800.0
23400.0
45000.0
46800.0

It has been found that memory limits can be reached when applying a small output increment over a long time period on the **OS** card. If AdH fails with a memory limit at this location when reading the boundary condition file, modifying the total number of output times in the output series is likely to fix the problem.

OS

AUTO-BUILD OUTPUT SERIES

Field	Type	Value	Description
1	char	OS	Card type
2	int	> 0	ID number of the series
3	int	> 0	Number of points in the series
4	int	> 0	Output units (Units 0 = seconds, 1 = minutes, 2 = hours, 3 = days, 4 = weeks)

4.10.3 Standard screen output

The standard screen output is in a tab delimited format. However, if the user chooses, additional formats of the standard output can be obtained by including a **PC LVL** card in the boundary condition file. The screen output is always given in seconds.

The first column of data gives the physics being solved for that iteration. For hydrodynamics the physics is listed as **HYD,** and for transport it is listed as **TRN** or **SLT** (for salinity). The order of the data in the default format from left to right is:

- physics
- time
- timestep size
- percent completion progress
- nonlinear iteration number
- linear iteration count
- node count after adaption
- failure flag (#)

The order of the data in the long column tabular form (**PC LVL** 1) from left to right is:

- physics
- time
- timestep size
- percent completion progress
- nonlinear iteration number
- linear iteration count
- maximum residual norm
- node number giving the maximum (worst) residual
- x, y, and z-coordinates of this worst node
- maximum solution increment norm,
- node number giving this maximum (worst) increment
- x, y, and z-coordinates of this worst node
- node count after adaption
- failure flag (#)

All time values in the screen output are in seconds. The maximum nonlinear residual is used to determine convergence against the **NTL** value, if included in the boundary condition file. The maximum solution increment is used to determine convergence against the **ITL** value, if included in the boundary condition file. If no adaption is taking place, the node count after adaption will not change throughout the run. Failure is specified as "#" and indicates that convergence did not occur and the timestep will be cut to ¼ the previous value. This column is left empty in all other instances. When adapting, "-2" in the screen output file indicates an added node. For transport iterations, the same information is provided with the **TRN** or **SLT** indicator in the first column.

PC LVL

SCREEN OUTPUT FORMAT

Field	Type	Value	Description
1	char	PC	Card type
2	char	LVL	Screen output format – level 0 is default
3	int	0, 1, 2	0 gives short column format; 1 gives long column format; 2 gives original AdH format (non-tabular)

4.10.4 Solution output files

Basic hydrodynamic results for velocity and depth are provided for all simulations. The depth files are stored as *filename*_dep.dat and the velocity data files are stored as *filename*_ovl.dat. Two files are provided to assist with setting parameters for adaption in AdH. These are *filename*_error.dat and *filename*_error_hydro.dat. The data in these files are indicators of where more mesh resolution may be able to improve the solution. It is not an indicator of the level of accuracy of the model as it has no field data for comparison. The *filename*_error_hydro.dat includes the adaption indicators for hydro only. The *filename*_error.dat includes the larger of the error values from hydrodynamics and transport for each node, normalized by the adaption tolerance values (**SRT** and/or **TRT**) in the boundary condition file.

All of these *.dat files are in SMS and GMS standard format, with a value for every node in the mesh at each user specified time.

4.10.5 Flux Output

Another output option is a calculation of flow and constituent mass across strings. The calculated flow across edge-strings (**EGS**) and mid-strings (**MDS**) is output at every time step when the **FLX** card is included.

Strings are selected for flow output with the **FLX** card, followed by the string number. The strings can be existing edge-strings (**EGS**) used for boundary condition input or mid-strings (**MDS**) created specifically for flow output (if the string exists for a boundary condition, do not recreate it, you cannot have duplicate string segments in AdH).

The **MDS** card has the same format as the **EGS** card. Strings used for flow output should begin and end on a mesh boundary or on a node that never wets and strings cannot have common elements shared between them.

Values will be output into the *filename*_tflx file. This file format is a table of values giving time, string #, flux value, and water surface values. Therefore, results for all requested strings will be in the same file. This file is not an output file read by SMS, GMS, or CMB. When simulating transport of any kind, the concentration flux is also saved with the inclusion of the FLX card. The results of these computations for all transported constituents are in the *filename*_conflx file. The example below indicates that flux output will be provided for strings 1 and 2.

```
FLX 1
FLX 2
```

FLX

FLOW OUTPUT

Field	Type	Value	Description
1	char	FLX	Parameter
2	int	> 0	String ID number for the mid string or edge string for which flow is to be output

4.10.6 Output Adapted Mesh and Solutions

When including adaption in the simulation it may be desired to see the adapted meshes. To request that the meshes are saved with the associated hydrodynamic solution files, the **PC ADP** card can be included in the boundary condition file. By including this card, the mesh and associated solution files will be saved at the time step intervals specified on the output control card. The output files will be named like so: *filename*.3dm-*timestep#*.0, *filename*.dep-*timestep#*.0, *filename*.ovl-*timestep#*.0 which is a geometry file for each timestep, the depths for each timestep, and the velocities for each timestep.

PC ADP

ADAPTED MESH PRINTING

Field	Type	Value	Description
1	char	PC	Card type
2	char	ADP	Adaptive mesh printing turned on - omit to turn off

4.10.7 MEO Output

AdH can output the mass error to the screen. To get the mass error output to the screen from AdH, the user must include the **PC MEO** card and set it to 1. This will enable the screen output of the mass error. By default this output is turned off.

PC MEO

MASS ERROR OUTPUT PRINT CONTROL

Field	Type	Value	Description
1	char	PC	Card type
2	char	MEO	MEO output from AdH simulation
3	int	0, 1	0 is the default (does not print), 1 prints the mass error output to screen

4.11 End of Boundary Condition File ⇧

An **END** statement is required at the end of the boundary condition file. The code will read the boundary condition file through to the **END** statement. Any information in the boundary condition file after the **END** statement will not be read as input to the simulation.

END

STOPPING THE MODEL

Field	Type	Value	Description
1	char	END	Close the model

5 Optional Features in AdH

The previous chapters of this guide define several options and cards necessary to run 2D-SW AdH with specific physics included. Most of these cards are required and therefore necessary for all simulations. This chapter will focus on how to include in a boundary condition file some of the most frequently used, optional physics.

5.1 Include Wind in the Simulation

Wind can be included at a single point, or several points, or through a wind field similar to ADCIRC. The card combination will determine the wind options being implemented.

5.1.1 Wind input options

5.1.1.1 Stress input, Wu, and Teeter methods

There are three methods that are currently included for wind stress calculations in AdH. The first is by providing the wind stress components directly in the boundary condition file, i.e. not applying any transformation to the supplied values. The second is the Wu method, which is typically used for deep waters, while the third, the Teeter method, is used for very shallow water. The Wu and Teeter methods require wind velocities be provided in the boundary condition file. The default calculation of wind shear assumes wind velocities are given and utilizes the Wu transformation. Alternative formulations for different materials within the model domain can be used by including the **MP WND STR** control card followed by the material number and a flag indicating the stress formulation to use.

The Wu and Teeter methods determine the wind stress in the x- and y-directions from the wind speed and direction by the following formulation.

$$T_{wx} = \rho_a C W^2 \cos\theta$$

$$T_{wy} = \rho_a C W^2 \sin\theta$$

Where ρ_a = air density, W = wind speed in L/T (m/s, ft/s) at 10 meter height, Θ = wind direction measured counter-clockwise from East. C is a wind stress coefficient defined according to the method used.

The equation below is for the Wu method (Wu, 1982) where W is the magnitude of the wind speed in m/s.

$$C = 0.001(0.8 + 0.065W)$$

The Teeter method for computing wind shear stress gives a wind stress coefficient based on water depth and wind speed. W is the wind speed in meters per second and W^* is the wind speed limited to a minimum value of 5 m/s. The depth given by h and h^* is a limited depth set to a minimum value of 2.001 meters. For more specifics on the development of the Teeter method for wind shear, see Teeter 2002.

$$C = \left(\frac{0.4}{16.11 - 0.5\ln h - 2.48\ln W}\right)^2 \left[1.0 - \left(\frac{1.118}{\sqrt{W^*}}\right)\right] e^{-0.6(h^* - 2.0)}$$

Typically, AdH includes wind stresses (or velocities) by user-specification of some number of wind stations within the boundary condition file. When applying wind data to the model run, the wind stresses/velocities for each of the wind stations are specified as a time series on an **XY2** card. The x and y coordinates of the wind station are given on an **XYC** card with the same series number as the **XY2** card that provides the values for that location.

The wind components are then provided for the station. The wind terms are given using the wind series card and have two values at each time given (the x- and y-component of the wind term). These values are in units consistent with the model units, and AdH will convert them if necessary for the wind stress coefficient equations to be accurate. An example using both the Wu and Teeter transforms (two materials) is as follows such that material 1 will use the Wu method and material 2 will use the Teeter method. The station is located at the coordinate 5.0, 15.0, and the values (wind velocities since Wu and Teeter are selected) over time are given in series 4 for 0 to 2000 seconds.

```
MP WND STR 1 1
MP WND STR 2 2
```

XYC 4 5.0 15.0
XY2 4 3 0 0
0.0 0.1 0.1
100.0 0.25 0.2
2000.0 0.25 0.2

MP WND STR

WIND STRESS

Field	Type	Value	Description
1	char	MP	Card type
2	char	WND	Parameter
3	char	STR	Parameter
4	int	≥ 1	Material type ID number
5	real	0,1,2	Wind transform (0 = no transform 1 = Wu 2 = Teeter)

XY2

X-Y-Y SERIES

Field	Type	Value	Description
1	char	XY2	Card type
2	int	> 0	ID number of the series
3	int	> 0	Number of points in the series
4	int	> 0	Input units. (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)
5	int	> 0	Output units (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)

Currently, only the data that is to be used for wind series is to be input via the X-Y-Y series.

XYC

WIND STATION COORDINATES

Field	Type	Value	Description
1	char	XYC	Card type
2	int	≥ 1	ID number of the series to which it is associated
3	real	#	X coordinate of the wind station
4	real	#	Y coordinate of the wind station

5.1.2 Wind attenuation

Wind attenuation is a means to increase or decrease the wind shear stress magnitude by applying a scale factor. This option is useful for manipulating wind values that may be measured over open water but are being applied in vegetated or populated areas. The wind attenuation is a material property card, **MP WND ATT**, and is followed by the material number and the factor to apply to the supplied wind speeds or shears. The default is 1.0 such that the wind is applied fully. As the fraction decreases, less of the wind stress is applied for the specified material region and vice versa.

MP WND ATT

WIND ATTENUATION

Field	Type	Value	Description
1	char	MP	Card type
2	char	WND	Parameter
3	char	ATT	Parameter
4	int	≥ 1	Material type ID number
5	real	0 ≤ #	Fraction applied to wind stress (default is 1.0)

5.2 Include Waves in the Simulation

AdH can include the effects of short waves in a simulation in order to show the impact of these wave forces on the velocities and depths. These parameters can be supplied as initial conditions or by coupling of other wave models, specifically STWAVE, via a tool such as CSTORM-MS (http://www.erdc.usace.army.mil/).

To tell AdH that waves will be supplied in the model, the **OP WAV** card is added as an operation parameter. This card tells AdH to look for wave data in the hotstart file. The wave stress must be provided to run AdH with waves and all other wave parameters are optional. Specific names must be provided when supplying the wave data in the hotstart file: WAVE_STRESS, WAVE_PERIOD, WAVE_HEIGHT, WAVE_ANGLE, WAVE_NUMBER, WAVE_BREAK, WAVE_SPEED, WAVE_ENERGY_B, WAVE_ENERGY_F. The wave stress data must be ordered as S_{xx}, S_{yy}, S_{xy}.

OP WAV

SHORT WAVE STRESSING

Field	Type	Value	Description
1	char	OP	Card type
2	char	WAV	Parameter

5.3 Superconvergent Patch Revovery Method

The **OP NF2** operational parameter activates the computation of 2D shallow water gradients. When this card is included, two additional output files will be generated that contain the velocity gradients with time. The **OP TPG** card can be included to inform AdH of the coefficient to be utilized in Streamwise Upwind Petrov-Galerkin (SUPG) stabilization for convection dominated flows. These outputs are most often used for numerical fish simulation.

OP NF2

SW2 GRADIENTS

Field	Type	Value	Description
1	char	OP	Card type
2	char	NF2	Parameter

OP TPG

PETROV-GALERKIN COEFFICIENT

Field	Type	Value	Description
1	char	OP	Card type
2	char	TPG	Parameter
3	real	$0.5 \geq \# \geq 0$	Coefficient for the Petrov-Galerkin equation

The **PC ELM** card can be included to activate the printout in TecPlot format of numerical fish surrogate information. If **PC ELM** is turned on the **OP NF2** card must be included in the boundary condition file. Omission will result in catastrophic failure during AdH runtime.

PC ELM

NUMERICAL FISH SURROGATE OUTPUT

Field	Type	Value	Description
1	char	PC	Card type
2	char	ELM	Numerical Fish Surrogate output (optional card)

5.4 Simulating vessels in a waterway

AdH has the ability to simulate the presence of vessels moving within a waterway. This is accomplished by calculating a pressure field, which applies a draft equal to that of the modeled vessel. The vessel characteristics are specified in a boat definition file (*filename*.bt), which will be read by Pre_AdH if the **OP BT** card is given in the boundary condition file.

Also, bed shear stresses due to vessel entrainment will be calculated and included as output upon inclusion of the **OP BTS** card. Use of this card requires inclusion of an additional card in the boat definition file (**PROP**).

NOTE: There can be no blank lines in the boat definition file.

For additional information, see Hammack et al. (2008) and Hammack and Tate (2008) as well as the AdH website.

OP BT

VESSEL MOVEMENT LIBRARY INCLUSION (ENABLE VESSEL MOVEMENT)

Field	Type	Value	Description
1	char	OP	Card type
2	char	BT	Parameter

OP BTS

ENABLE VESSEL ENTRAINMENT

Field	Type	Value	Description
1	char	OP	Card type
2	char	BTS	Parameter

5.5 Include Constituents in the Simulation

After finding a flow solution, an associated transport problem can be solved. To add any transported constituent, the **OP TRN** card should be set for the number of quantities being included. Every transported constituent will also require the necessary transport constituent properties discussed below. An error message will be displayed if transport properties are included in the input file but no transport quantities have been specified on the **OP TRN** card.

The following card specifies one transported quantity:

OP TRN 1

The constituent cards are provided so that specific transported quantities and sediment types can be accounted for. There is a separate card for each constituent type. Vorticity (**CN VOR**), temperature (**CN TMP**), and salinity (**CN SAL**) are transported as constituents. Other constituent types can be categorized as a general constituent, such as dye tracers or passive tracers with no density, and use the **CN CON** card. Also included on the constituent cards are the constituent ID number, the reference concentration, and other constituent specific parameters.

CN CON 1 10.0 ! generic constituent

The first number indicates the constituent number. The next number for all cases is the reference concentration. The reference concentration is used to normalize the concentration values. This is done so that the loss of information in the internal computations associated with the truncation of significant figures is minimized. The value of the reference concentration should be a typical value that one might expect to observe in the prototype being modeled, for example if during run time the typical average value of a constituent is expected to be 35 the reference concentration should be specified at 35. Doing so will scale all expected values of this constituent to be around 1 in the internal computations, not in the output files. If uncertain about how to set this parameter, it should be set equal to 1.0. The results for each constituent are saved according to the user specified options in a file entitled *filename_con#.dat* where # is the constituent ID number.

OP TRN

TRANSPORT EQUATIONS

Field	Type	Value	Description
1	char	OP	Card type
2	char	TRN	Parameter
3	int	≥ 0	Total number of transported materials

CN CON

ANY GENERIC CONSERVATIVE CONSTITUENT

Field	Type	Value	Description
1	char	CN	Card type
2	char	CON	Parameter
3	int	≥ 1	Constituent ID number
4	real	> 0	Characteristic concentration

5.5.1 Baroclinic transport (Salinity and Temperature)

AdH typically computes independent of density, meaning that the solutions are not affected by density gradients. However, salinity and temperature affect density and must be simulated as such. AdH will automatically invoke the baroclinic effects so that these transport constituents can be model appropriately when **CN SAL** and **CN TMP** cards are used.

Here is an example of salinity transport using a reference concentration of 35. Salinity is computed in parts per thousand (ppt).

CN SAL 1 35.0

Specification of the **CN TMP** card activates the transport of heat in AdH. Below are examples of **CN TMP** card usage. Temperature is computed in Celsius.

To activate heat transport **without** heat transfer through the air/water interface the card below is specified

CN TMP 1 1.0 0

To activate heat transport **with** heat transfer through the air/water interface the card below is specified

CN TMP 1 1.0 1

If heat transfer through the air/water interface is activated the user must specify the **DB RAD** card as well. The concept of equilibrium transport is used to simulate the rise and fall of temperature as a response to short wave radiation as well as the dew point temperature.

CN SAL

SALINITY (BAROCLINIC TRANSPORT)

Field	Type	Value	Description
1	char	CN	Card type
2	char	SAL	Parameter
3	int	≥ 1	The constituent ID number
4	real	> 0	Reference concentration (ppt)

CN TMP

TEMPERATURE (BAROCLINIC TRANSPORT)

Field	Type	Value	Description
1	char	CN	Card type
2	char	TMP	Parameter
3	int	≥ 1	The constituent ID number
4	real	> 0	Reference concentration (Celsius)
5	int	1, 2	Heat transfer though air/water interface (0 = no, 1 = yes)

DB RAD

DIRICHLET – SHORT WAVE RADIATION AND DEW POINT TEMPERATURE

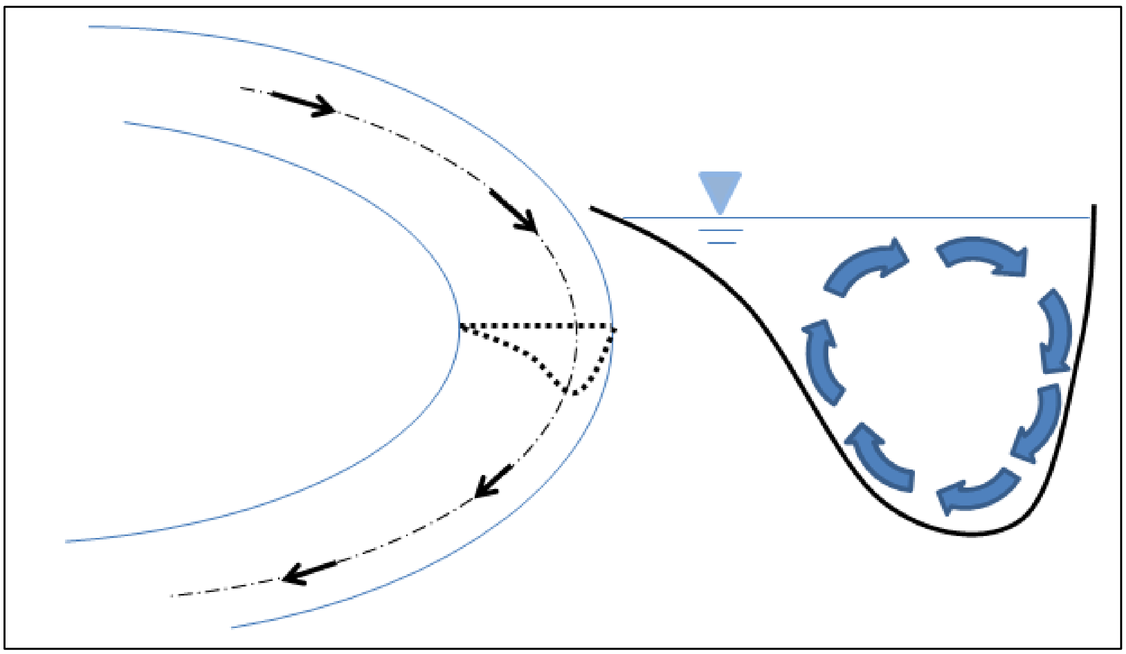
Field	Type	Value	Description
1	char	DB	Card type
2	char	RAD	Parameter
3	int	≥ 1	Series ID number with short wave radiation in w/m^2
4	int	≥ 1	Series ID number that dew point temperature in Celsius

5.5.2 Vorticity transport-bendway correction

A method for correcting 2-D models for the 3-D effects of vorticity around bends has been included in AdH. The development of the method is given by Bernard (1992). Vorticity is activated voluntarily with the **CN VOR** card. The vorticity is included as a transport constituent due to its constituent-like behavior as it moves within the model and must therefore be included on the **OP TRN** card. The units of the vorticity computed in AdH are radians/time.

Vorticity here refers to the helical flow that is generated by the angular momentum of the flow in a bendway. Since the surface water moves faster than the water near the bed, the surface water is preferentially transported to the outside of the bend and the bottom water transports to the inside (due to conservation of water mass). This mechanism is what drives the meandering process of rivers and point bar formation. Figure 10 illustrates this process.

Figure 10. Illustration of helical flow in river bends



Although the depth-averaged flow direction is not greatly influenced by this helical flow (again, due to conservation of water mass) the helical flow induces a momentum flux to the outside of the bend. This momentum flux varies spatially and temporally, as the flow varies. The vorticity concentration, then, is the correction term associated with this momentum flux. It is used to scale a transverse momentum flux term that is included in the model equations.

Including **CN VOR** in the boundary condition file enables the bendway correction. This card is followed by the constituent ID number, a normalization factor, A_s term, and D_s term. The A_s and D_s terms are semi-empirical coefficients determined by comparisons against measured values. AdH uses default values of A_s = 5.0 and D_s = 0.5 which will be set automatically if these terms are input as 0.0 in the boundary condition file. Changing these defaults is not recommended unless careful consideration and review of the reference material (Bernard 1992) is made.

CN VOR 1 1.0 0.0 0.0

However, it is possible to not want to include vorticity transport in all areas of the model domain, specifically in vegetated or wetland areas where the shallow flows may… . Vorticity can be essentially turned off by including an **MP NVM** card followed by the material number where you want to remove the vorticity impact.

CN VOR

VORTICITY TRANSPORT - BENDWAY CORRECTION

Field	Type	Value	Description
1	char	CN	Card type
2	char	VOR	Parameter
3	int	≥ 1	The constituent ID number
4	real	> 0	Normalization factor
5	real	≥ 0	A_s term, default is 0.0 which sets A_s = 5.0
6	real	≥ 0	D_s term, default is 0.0 which sets D_s = 0.5

MP NVM

NO VORTICITY TRANSPORT BY MATERIAL

Field	Type	Value	Description
1	char	MP	Card type
2	char	NVM	Parameter
3	int	≥ 1	The material ID number

5.5.3 Constituent diffusion

When any of the available transport constituents are included in the model a turbulent diffusion card can be included to define the degree of spreading of the constituent. The diffusion rate must be specified for each material and constituent. This diffusion card is required when using the constant eddy viscosity (**MP EVS)** option. When using the estimated eddy viscosity (**MP EEV)** option the diffusion is equal to the eddy viscosity computed using the **MP EEV** card. So, for a given material type, any **DF** card will be ignored if the **EEV** card is active.

MP DF

TURBULENT DIFFUSION RATE

Field	Type	Value	Description
1	char	MP	Card type
2	char	DF	Parameter
3	int	≥ 1	Material type ID number
4	int	≥ 0	Constituent ID number
5	real	≥ 0.0	Turbulent diffusion rate

5.5.4 Constituent solution controls

Constituent concentrations associated with inflowing water at boundary conditions must be defined with boundary conditions. Constituent concentration can be applied as a natural boundary condition (**NB**) on an edge string or as a Dirichlet (strongly enforced, **DB**) boundary condition along a node string.

NB TRN

NATURAL BOUNDARY CONDITION - TRANSPORT

Field	Type	Value	Description
1	char	NB	Card type
2	char	TRN	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Constituent ID number
5	int	≥ 1	Series ID number that contains the constituent concentration (units dependent of the transport type)

DB TRN

DIRICHLET BOUNDARY CONDITION - TRANSPORT

Field	Type	Value	Description
1	char	NB	Card type
2	char	TRN	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Constituent ID number
5	int	≥ 1	Series ID number that contains the constituent concentration (units dependent of the transport type)

5.5.5 Constituent output

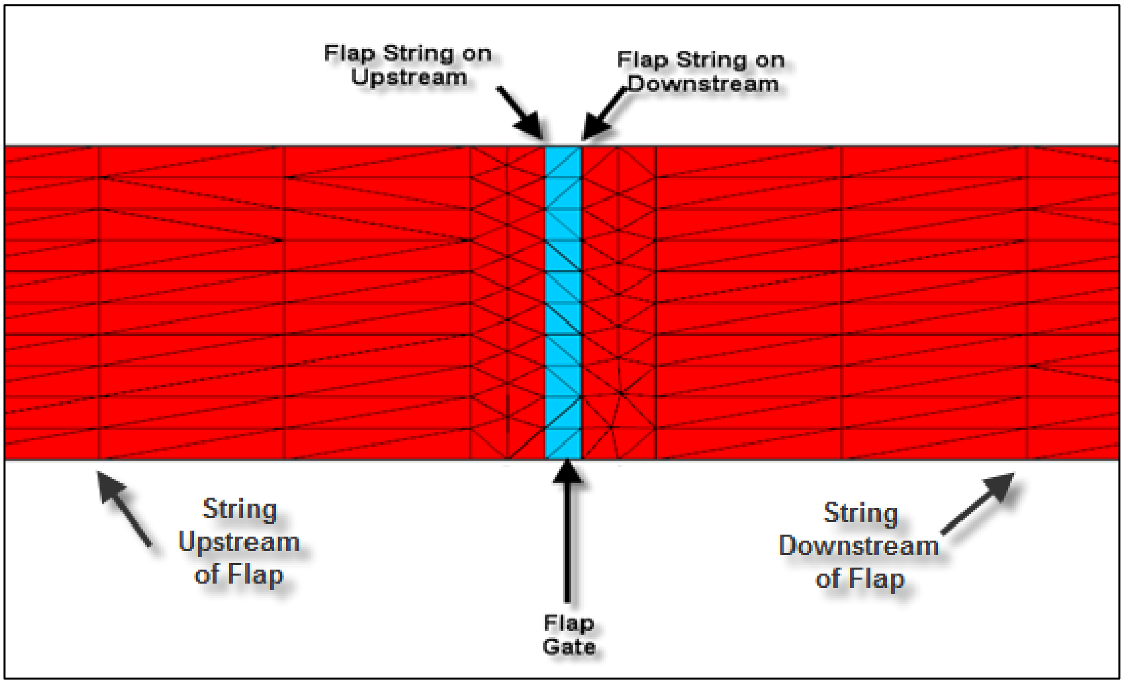
The constituent results are stored in the form of concentration values and named *filename*_con#.dat, where # is the transport constituent number. The unit of the solution is dependent on the transport type. When adapting the model mesh during the simulation, the concentration results can be stored for the adapted mesh by including the **PC ADP** card. A new file will be produced at every timestep for the adapted grid and all simulated constituents. The **PC ADP** card will provide all data sets and the mesh files, both hydro and transport.

5.6 Include hydraulic structures in the simulation

Several hydraulic structure options are available in AdH. These include flap gates, weirs, and sluice gates. The shallow water equations are hydrostatic, meaning that the vertical accelerations are ignored. When modeling hydraulic structures in a shallow water code, the focus is on accurate flow passage through the structure, not on the dynamics at the structure itself. A full Navier-Stokes equation set is necessary to simulate the details of the flow through the structure, which is a separate module of AdH.

A description of the necessary strings for all AdH structure types is given in Figure 11 below, simply replace "flap" with the appropriate structure type. The colored sections in the figure represent material definitions so that the structure itself can be turned **OFF** in the computations (the blue material string must be turned off). All structure options require four strings defined: two edge strings along the upstream and downstream sides of the structure (flap string) and two node strings several elements upstream and downstream of the structure used to compute water surface elevation difference.

Figure 11. Structure definitions for AdH input



5.6.1 Flap gates

Flap gates are defined according to user-supplied coefficients. The number of gates is given on the **FLP** card and the definition of each gate is given on the **FGT** card. Flap gate flow is computed using a user defined polynomial rating curve in the form:

$$Q = A(\Delta h)^B + C\Delta h + D$$

where

$$\Delta h = WSE_{U/S} - WSE_{D/S}$$

Six user-specified coefficients are available such that a larger polynomial expression can be implemented, but at this time only four are used and the additional values are simply placeholders for future modeling options. The flap gate must be specified by a separate material and then that material must be turned **OFF**. To do this, use the **OFF** card followed by the flap gate material string number.

FLP

NUMBER OF FLAP GATES

Field	Type	Value	Description
1	char	FLP	Card type
2	int	≥ 1	Number of flap gates

FGT

FLAP GATE PARAMETERS

Field	Type	Value	Description
1	char	FGT	Card type
2	int	≥ 1	Flap gate number
3	int	= 1	1 – User specified parameters
		= 2	2 – Automatic computation (not yet implemented)
4	int	≥ 1	String upstream of flap (node)
5	int	≥ 1	String downstream of flap (node)
6	int	≥ 1	Flap string on the upstream (edge)
7	int	≥ 1	Flap string on the downstream (edge)
8	real	≥ 0	Coefficient A
9	real	≥ 0	Coefficient B
10	real	≥ 0	Coefficient C
11	real	≥0	Coefficient D
12	real	≥0	Coefficient E
13	real	≥0	Coefficient F
14	real	≥0	Length of flap gate

5.6.2 Weirs

The **WRS** card is used in conjunction with the **WER** where the number of weirs being modeled is given and then the definition of the weir is provided. The equation for the flow across the weir (Q) is:

$$Q = CLFH^{\frac{3}{2}}$$

Where C is the weir coefficient for flow over the unsubmerged weir, L is the length of the weir, F is the correction coefficient for submergence of the weir, and H is the upstream head over the weir crest elevation given by

$$H = Z_{U/S} - Z_{weir}$$

A description of the weir implementation is provided in Savant and Berger (2009) and Figure 11 shows the definitions required (same for weir and flap gate).

The weir card can also be used to specify a free flowing weir as a model boundary condition. To do this, the downstream reference string is defined as the same string as the upstream reference string and the downstream weir string is defined as the same string as the upstream weir string.

The weir must be specified by a separate material and that material must be turned **OFF**. To do this, use the **OFF** card followed by the weir material string number.

WER

NUMBER OF WEIRS

Field	Type	Value	Description
1	char	WER	Card type
2	int	≥ 1	Number of weirs

WRS

WEIR PARAMETERS

Field	Type	Value	Description
1	char	WRS	Card type
2	int	≥ 1	Weir Number
3	int	≥ 1	String upstream of weir (node)
4	int	≥ 1	String downstream of weir (node)
5	int	≥ 1	Weir string on upstream (edge)
6	int	≥ 1	Weir string on downstream (edge)
7	real	≥ 0	Length of weir
8	real	≥ 0	Weir crest elevation
9	real	≥ 0	Weir height

5.6.3 Sluice gates

Sluice gates are included in the same manner as flap gates and weirs. The dynamics at the structure itself are not modeled but the correct passage of flow is computed based on user-supplied coefficients. The sluice gate equations used in AdH for free flow and submerged flow are based on work by Swamee (1992) and given below.

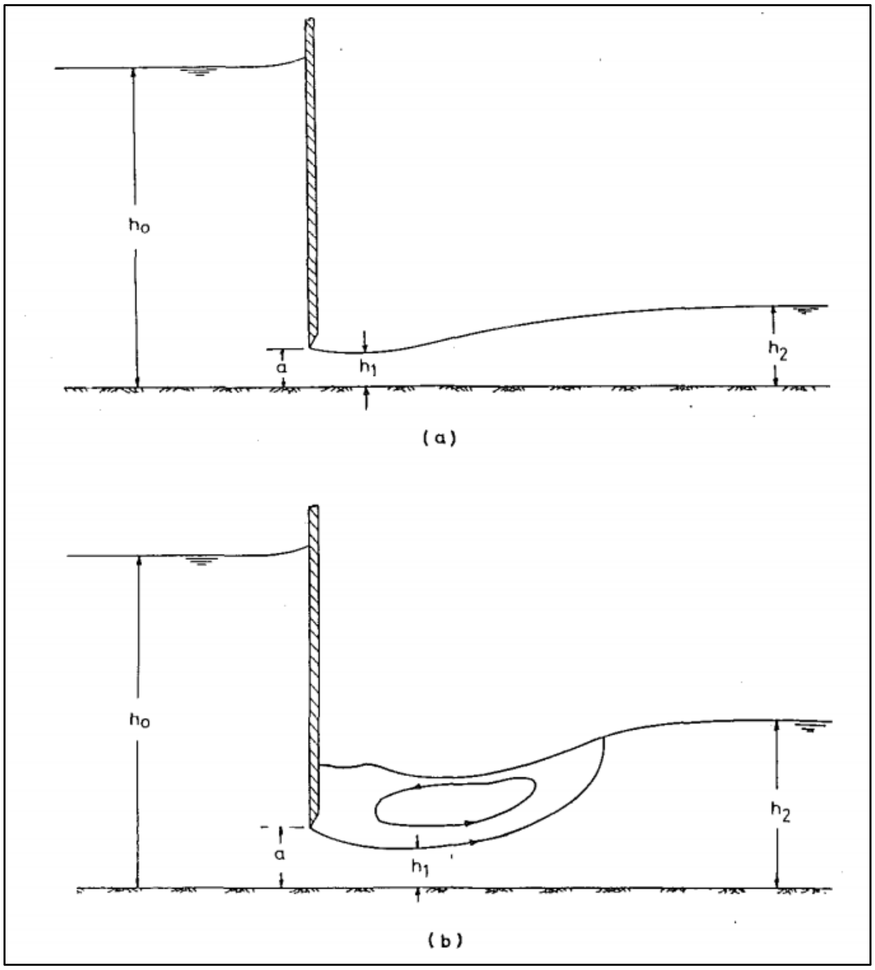
$$Q = C_d ab\sqrt{2gh_0}$$

$$Q = \frac{0.864ab\sqrt{gh_0}\left(\frac{h_0 - a}{h_0 + 15a}\right)^{0.072}(h_0 - h_2)^{0.7}}{0.32\left[0.81h_2\left(\frac{h_2}{a}\right)^{0.72} - h_0\right]^{0.7} + (h_0 - h_2)^{0.7}}$$

Where a is the sluice gate opening, b is the sluice gate length, g is gravity, h_o is the upstream depth, h_2 is the tailwater depth, and Q is the discharge through the structure. Figure 12 shows these parameters.

The **SLUICE** card gives the number of sluice gates included in the model. The **SLS** card lists the necessary sluice function parameters for each gate. Unlike other structure types, the sluice gate requires an **XY** time series to define the sluice gate opening, “a” in the equations above and the figure below. The sluice gate must be specified by a separate material and then that material must be turned **OFF**. To do this, use the **OFF** card followed by the sluice gate material string number.

Figure 12. Sluice gate definitions; (a) free flow, (b) submerged flow (from Swamee 1992)



SLUICE

NUMBER OF SLUICE GATES

Field	Type	Value	Description
1	char	SLUICE	Card type
2	int	≥ 1	Number of sluices

SLS

SLUICE GATE PARAMETERS

Field	Type	Value	Description
1	char	SLS	Card type
2	int	≥ 1	Sluice Gate Number
3	int	≥ 1	String upstream of sluice gate (node)
4	int	≥ 1	String downstream of sluice gate (node)
5	int	≥ 1	Sluice string on upstream (edge)
6	int	≥ 1	Sluice string on downstream (edge)
7	real	≥ 0	Length of sluice gate (*b* in the previous equations)
8	int	≥ 1	Time series defining the sluice gate opening over time (*a* in the previous equations)

5.7 Breach Library

In addition to typical structures that are often included in modeling, a breaching library is included in AdH and allows for several boundary condition options. These cards are intended to assist engineers with realistic dam and levee breach simulations by allowing users to input time varying bed elevations to define the breach displacement or use one of several documented formulations. Seven breach formulations are available in AdH along with a basic option to define the breach elevation over time. The seven formulations are:

- Johnson and Illes (1976) formulation, suitable for earth, gravity, and arch concrete dams
- Singh and Snorrason (1982, 1974) formulation, suitable for earthen dams
- MacDonald and Langridge-Monopolis (1984) formulation, suitable for earthfill and non-earthfill dams
- Froelich (1987, 1995) formulation, suitable for engineered earthen or slag dams
- Bureau of Reclamation (1988) formulation, suitable for earthen dams
- Von Thun and Gillette (1990) formulation, suitable for engineered dams with or without clay cores
- Federal Energy Regulatory Commission, FERC (1987) formulation, suitable for engineered and nonengineered earthen and slag dams.

All of the breach boundary condition options are dirichlet conditions and require a **BR** card. The particular formulation desired follows the **BR** card along with any necessary parameters. The most basic breach option is the **USR** (user supplied) condition such that values are assigned at particular nodes and can change over time. This condition does not apply any functions - it simply moves the bed at a node according to the given elevation. The seven included library formulations are given by **JAI**, **SAS**, **MLM**, **FRO**, **BRC**, **VTG**, and **FER**. The specific card structures will be given in the table at the end of this section and details are given in Savant (2013).

An additional output file is supplied when using a breach boundary condition card. The *filename*_belev.dat file gives the change in the bed elevation throughout the simulation in order to track the breach conditions.

BR JAI

DIRICHLET – BREACH JOHNSON AND ILLES

Field	Type	Value	Description
1	char	BR	Card type
2	char	JAI	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≥ 0	Breach failure time (seconds)
9	int	≥ 1	Side slope node furthest from breach (if side slope section)
10	int	≥ 1	Side slope node closest to breach (if side slope section)

BR SAS

DIRICHLET – BREACH SINGH AND SNORRASON

Field	Type	Value	Description
1	char	BR	Card type
2	char	SAS	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0.0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≤ 3600	Breach failure time (seconds - complete failure by 3600s)
9	int	≥ 1	Side slope node furthest from breach (if side slope section)
10	int	≥ 1	Side slope node closest to breach (if side slope section)

BR MLM

DIRICHLET – BREACH MACDONALD AND LANDGRIDGE-MONOPOLIS

Field	Type	Value	Description
1	char	BR	Card type
2	char	MLM	Parameter
5	real	≥ 0.0	Maximum water depth above breach bottom (meters)
6	real	≥ 0.0	Reservoir volume (cubic meters)
7	real	#	Minimum breach elevation (meters)
8	real	#	Dam/Levee crest elevation (meters)

BR FRO

DIRICHLET – BREACH FROELICH

Field	Type	Value	Description
1	char	BR	Card type
2	char	FRO	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0.0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≥ 0.0	Breach failure time (seconds)
9	real	0.1≥#≤4.0	Exponent for main breach, controls the rate of the main breach progression
10	int	≥ 1	Side slope node furthest from breach (if side slope section)
11	int	≥ 1	Side slope node closest to breach (if side slope section)

BR BRC

DIRICHLET – BREACH BUREAU OF RECLAMATION

Field	Type	Value	Description
1	char	BR	Card type
2	char	BRC	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≥ 0	Breach failure time (seconds)
9	int	≥ 1	Side slope node furthest from breach (if side slope section)
10	int	≥ 1	Side slope node closest to breach (if side slope section)

BR VTG

DIRICHLET – BREACH VON THUN AND GILLETTE

Field	Type	Value	Description
1	char	BR	Card type
2	char	VTG	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	int	0 or 1	Erosion character – 0=easily erodible, 1=erosion resistant
9	real	≥ 0	Breach failure time (seconds)
10	int	≥ 1	Side slope node furthest from breach (if side slope section)
11	int	≥ 1	Side slope node closest to breach (if side slope section)

BR FER

DIRICHLET – BREACH FEDERAL ENERGY REGULATORY COMMISSION

Field	Type	Value	Description
1	char	BR	Card type
2	char	FER	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	int	0 or 1	Engineer character – 0=non-engineered, 1=engineered/well compacted
9	real	≥ 0	Breach failure time (seconds)
10	int	≥ 1	Side slope node furthest from breach (if side slope section)
11	int	≥ 1	Side slope node closest to breach (if side slope section)

BR USR

DIRICHLET – USER DEFINED BREACH DISPLACEMENT

Field	Type	Value	Description
1	char	BR	Card type
2	char	USR	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the elevation of the string

5.8 Tidal Constituent Boundary

AdH can take in tidal constituent information and generate an XY series for the boundary condition water level. The boundary condition must have an **NB TID** card for the solution control with the number of the edge string where it should be applied. Two additional files are required, one defining the tidal constituents (*.tides) and one defining the yearly components for the tide (*.tides_yearly). The constituent file contains a header row followed by a row of data defining the number of constituents to be read, the number of boundary condition time series to generate, a time shift from January 1, a datum shift from MTL, the starting time in seconds, the ending time in seconds, and the time increment for the series in seconds. The constituents must be listed in the NOAA order with a line for the constituent name followed by a line giving the boundary edge string number, the amplitude, and the phase angle for each boundary condition series. Here is an example showing constituent data for two boundary strings, 3 and 4. The series is generated over 1 day with a data point every half hour and no time or datum shift.

```
ncon  nbc   tshift   datum    t0       tf        dt
40    2     0.0      0.0      0.0      86400.0   1800.0
M2
3   0.945   264.2
4   0.400   264.2
S2
3   0.232   295.5
4   0.232   295.5
…
X1
3  0.000  0.0
4  0.000  0.0
```

The yearly data file includes nodal factors and equilibrium arguments such that the correct position of the tide is matched with the correct time. The yearly data for the year you are generating should be the only information in the *.tides_yearly file. Due to the formatted read for this file, all spaces in the equilibrium constituents are filled with zeroes. These yearly components can be computed based on Schureman (1941). Here is an example of the yearly data file for 2012.

```
2012 1 110171942100000001017041909600185103402850935171610522227097621 27   1   1
```

2012 1 11017194210000000101704190960018510340285093517161052222709762127 1 1

2012 1 21000000010342362101717891000000010170266101724960790053610170296 1 1

2012 1 31000180011232313094917491058152310002001100028011017165808590310 1 1

2012 1 40935156309350193100000311000176909352270100034991017165810262914 1 1

2012 1 51164157809930100088721651070057010171942087435331017204309510073 1 1

The equation for the time series tidal elevation based on the tidal constituents and yearly data is shown here.

$$\eta(x,y,t) = \sum A_i(x,y) f_i(t_0) cos\left[\frac{2\pi}{T_i}(t - t_o) + V_i(t_0) - \psi_i(x,y)\right]$$

$A_i = amplitude\ of\ constituent\ i$

$T_i = period, \quad \psi_i = phase$

$f_i = nodal\ correction\ factor, \ V_i = equilibrium\ argument$

NB TID

TIDAL CONSTITUENT BOUNDARY CONDITION

Field	**Type**	**Value**	**Description**
1	char	NB	Card type
2	char	TID	Parameter
3	int	≥ 1	String ID number (edge)

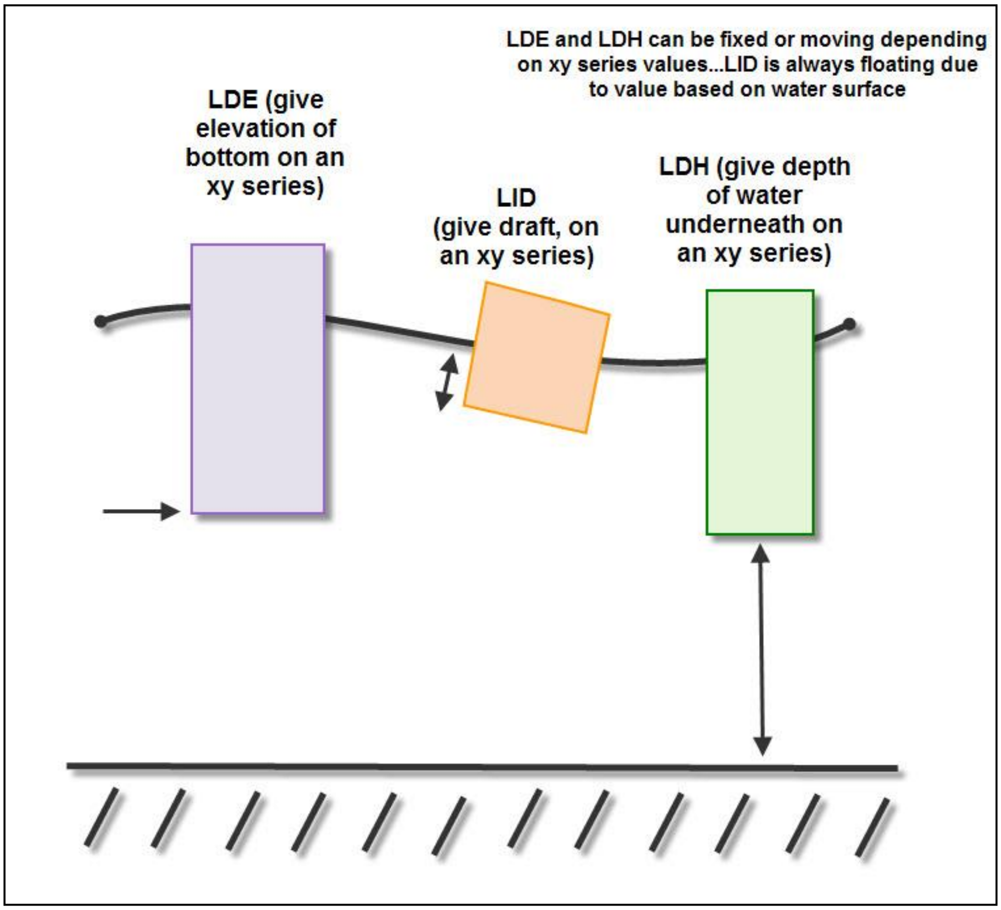
5.9 Stationary Lid in the Flow

If a vessel is moving in the waterway then the vessel movement library can move this pressure throughout the domain to represent the long-wave impacts on the waterway. If, however, a pressure field is stationary then it shouldn't be necessary to define a boat path and speed. For this case and the case in which a lid is prescribed in the flow we have developed another approach.

This method is implemented by selecting all the nodes that are to comprise the lid or pressure field and assigning them to a node string. This node string will then be assigned the lid elevation with a **DB LDE** card, the depth of the water with a **DB LDH** card, or the pressure (in terms of draft) that is desired with a **DB**

LID card. None of these parameters affect the friction that is applied. Also, since the depth or elevation is enforced via a penalty it will not be exact. The figure below shows the stationary lid options in the vertical plane.

Figure 13. Stationary lid definitions.



DB LDE

DIRICHLET - STATIONARY LID ELEVATION

Field	Type	Value	Description
1	char	DB	Card type
2	char	LDE	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the elevation to be implemented

DB LDH

DIRICHLET - DEPTH OF WATER UNDER STATIONARY LID

Field	Type	Value	Description
1	char	DB	Card type
2	char	LDH	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the depth

DB LID

DIRICHLET - FLOATING STATIONARY OBJECT

Field	Type	Value	Description
1	char	DB	Card type
2	char	LID	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the draft of the lid

6 Running AdH

Once the required input files have been created, the AdH model can be run. The commands for serial application are:

pre_adh *filename*
adh *filename*

where *filename* is the root of the model's filenames, i.e. for a model named pl8_AdH the following three files would be required for 2D simulation - pl8_AdH.3dm, pl8_AdH.hot and pl8_AdH.bc. All files must have the same *filename* as their root followed by the appropriate suffix. "pre_adh" and "adh" should be replaced with the actual executable name for the version of AdH you are running.

Pre_adh acts as an initial model setup and checks that the necessary input files do not have major problems. This step does not check for every possible error so there is no guarantee that the model will run successfully, but it can help find many issues that may be present in the input files.

Pre_adh writes out a binary file, called *filename*.adh. This file contains all of the information provided in the .bc, .hot, and .3dm files. It is only this file that is read by AdH.

To run AdH in multiprocessor mode, an mpi executable must the available on the computer. For Windows, the Microsoft HPC Pack must be installed. AdH has been tested with the 2008 and 2012 versions which are available through the Microsoft download center. Once installed, the multiprocessor compiled version of AdH can be executed on the PC with the command:

mpiexec –n # adh_mul filename

where *filename* is the root of the model's filenames and "adh_mul" is the executable name for the multiprocessor compiled version for the Windows PC.

After the AdH model is run, GMS, SMS, Paraview or other visualization tools can be used to visually inspect the results. The solution files available to visualize are dependent on the file output options included in the boundary condition file and the features included in the model. Transport simulations will include concentration output.

Table 3. AdH output file names.

Output filename conventions (*.dat)	
*_dep.dat	overland head (scalar, depth)
*_ovl.dat	overland velocity (vector)
*_err.dat	normalized residual error for use in setting refinement parameters (scalar)
*_err_hydro.dat	non-normalized residual error for the hydrodynamics (scalar)
*_con#.dat	constituent concentration, # = constituent number (scalar, units depend on transport type)
*err_con#.dat	non-normalized residual error for the transport constituent (scalar)
*_belev.dat	breach bed elevation (scalar, length)
*_conflx	concentration flux across a string for each constituent, is included when the FLX card is used followed by the string number…this is NOT an SMS file
*_tflx	hydrodynamic flux across a string, is included when the FLX card is used followed by the string number…this is NOT an SMS file
*_ugd	u-velocity gradients when using OP NF2
*_vgd	v-velocity gradients when using OP NF2
*_ELAM	numerical fish surrogate format output when using OP NF2 and PC ELAM
*_waveHeight.dat	wave height when using OP WAV (scalar)
*_waveForces.dat	wave forces when using OP WAV (x-y vector)
*_waveStress.dat	wave stress when using OP WAV (xx, yy, xy)
*_str.dat	bed shear stress when using OP BTS (scalar)
*_pdp.dat	previous timestep depth when using OP TEM
*_pov.dat	previous timestep velocity when using OP TEM

As an example, for a hydrodynamic simulation with salinity, vorticity, and a generic constituent, the output files would be: (information in parenthesis gives names for the hotstart file)

*_dep.dat (ioh)	Depth value
*_ovl.dat (iov)	X_vel, Y_vel, Z_vel (Z_vel = 0 for 2D)
*_err.dat	Residual error (normalized)
*_err_hydro.dat	Hydro residual error (non-normalized)
*_con1.dat (icon 1)	Concentration 1 (salinity for this example)
*_con2.dat (icon 2)	Concentration 2 (vorticity for this example)
*_con3.dat (icon 3)	Concentration 3 (generic constituent for this example)
*_err_con1.dat	Transport residual error for transport 1 (non-normalized)
*_err_con2.dat	Transport residual error for transport 2 (non-normalized)
*_err_con3.dat	Transport residual error for transport 3 (non-normalized)

References

Bakker, Andre', (2012) "The Colorful Fluid Mixing Gallery". http://www.bakker.org/cfm.

Bernard, B., (1992) "Depth-Average Numerical Modeling for Curved Channels" Technical Report HL-92-9, U.S. Army Corps of Engineers, Engineering Research and Development Center, Vicksburg, MS.

Blumberg, A. F., B. Galperin, and D. J. O'Connor. 1992. Modeling vertical structure of open-channel flows. *Journal of Hydraulic Engineering* 118: 1119–1134.

Brown, Gary L. (2017) "Mass, Energy and Momentum Flux Analysis of a Bendway Weir" *Journal of Hydraulic Engineering*, ASCE, September, 2017.

Burchard, H. 2001. On the *q2l* equation by Mellor and Yamada (1974). Notes and correspondence. *Journal of Physical Oceanography* 31: 1377–1387.

Burchard, H., O. Petersen, and T. Rippeth. 1998. Comparing the performance of the Mellor-Yamada and the k-e two-equation turbulence models. *Journal of Geophysical Research* 103: 10543–10554.

Christensen, B. A., (1972) "Incipient Motion on Cohesionless Channel Banks", *Sedimentation*, H.W. Shen, editor, Fort Collins , CO.

Christensen, B. A., (1985) "Open Channel and Sheet Flow Over Flexible Roughness" Proceedings, 21st IAHR Congress, Melbourne, Australia , 19-23 August.

Chu, Chia-Ren, Chih-Jung Huang, Tso-Ren Wu, Chung-Yue Wang, "Numerical simulation of hydrodynamic loading on submerged rectangular bridge decks" The Seventh International Colloquium on Bluff Body Aerodynamics and Applications (BBAA7) Shanghai, China; September 2-6, 2012

Federal Energy Regulatory Commission, 1987, Engineering Guidelines for the Evaluation of Hydropower Projects, FERC 0119-1, Office of Hydropower Licensing, July 1987, 9 p

Fischer, H.B., E.J. List, R.C.Y. Koh, J. Imberger, and N.H. Brooks. 1979. Mixing in Inland and Coastal Waters. Academic Press, New York.

French, R. H., and S. C. McCutcheon. 1983. *Turbulent vertical momentum transfer in stratified environments*. Publication No. 41079. Las Vegas, NV: Desert Research Institute.

Froelich,David C.,1987 "Embankment-Dam Breach parameters," Hydraulic engineering Proceedings of the 1987 ASCE National conference on Hydraulic Engineering ,Williamsburg Virginia, August 3-7,1987,p570-575

Froelich, David C., 1995,"Peak Outflow from Breached Embankment Dam,"Journal of water Resources Planning and management, vol.121, no.1.

Hammack, E. A., Smith, D. S., and Stockstill, R. L. "Modeling vessel-generated currents and bed shear stresses," Technical Report ERDC/CHL TR-08-7, U.S. Army Engineer Research and Development Center, Vicksburg, MS, June 2008.

Hammack, E. A. and J. N. Tate. 2008. Considerations for modeling vessel-generated currents and bed shear stresses. Coastal and Hydraulics Engineering Technical Note, http://chl.erdc.usace.army.mil/chetn, ERDC/CHL CHETN-IX-17. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Heath, Ronald E., Gary L. Brown, Charles D. Little, Thad C. Pratt, Jay J Ratcliff (2015). "Old River Control Complex Sedimentation Investigation". USACE ERDC Technical Report, ERDC/CHL TR-15-8.

Henderson-Sellers, B. 1982. A simple formula for vertical eddy diffusion coefficients under conditions of non-neutral stability. *Journal of Geophysical Research* 57(C8): 5860–5864.

Jacobs, Jennifer M., and Wang, Min-Hui (2003) "Atmospheric Momentum Roughness Applied to Stage-Discharge Relationships in Flood Plains" ASCE Journal of Hydrologic Engineering, March/April (2003)8:2(99).

Johnson, F.A., and Illes, 1976,"A Classification of Dams Failures," International Water Power and Dam Construction, December 1976, p43-45

Kent, R. E., and D. W. Pritchard. 1957. A test of mixing length theories in a coastal plain estuary. *Journal of Marine Research* 18(1): 62–72.

Lilly, D.K (1967) “The Representation of Small-Scale Turbulence in Numerical Experiments”, Proc. IBM Scientific Computing Symposium on Environmental Sciences, pp 195-210.

Macdonald, Thomas C., and Jennifer Langridge- Monoopolis, 1984,”Breaching Characteristics of Dam Failures,” Journal of Hydraulic Engineering, vol.110, no.5, pp 567-586

Massey, T. Chris. 2014. “Locally constrained nodal connectivity refinement procedures for unstructured triangular finite element meshes.” *Engineering with Computers (2015)*, Vol. 31, pp 375-386.

Mellor, G. L., and T. Yamada. 1982. Development of a turbulence closure model for geophysical fluid problems. *Reviews of Geophysics and Space Science* 20(4): 851–875.

Munk, W. H., and E. R. Anderson. 1948. Notes on a theory of the thermocline. *Journal of Marine Research* 7: 276.

Press, William H, Teukolsky, Vetterling, William T, Flannery Brian P “Numerical Recipes in C, The Art of Scientific Computing” Cambridge University Press, Second Edition, 1992.

Pritchard, D. W. 1960. The movement and mixing of contaminants in tidal estuaries. In *Proceedings of the First International Conference on Waste Disposal in the Marine Environment*. University of California at Berkeley. New York: Pergamon Press.

Rodi, W. “Turbulence Models and their Applications in Hydraulics – A State of the Art Review”, International Association for Hydraulics Research, 1984.

Savant, G., Berger, C, McAlpin, T.O. and Tate, J.N. (2011). “Efficient Implicit Finite-Element Hydrodynamic Model for Dam and Levee breach.” Journal of Hydraulic Engineering, Vol. 137, No:9.

Savant, Gaurav (2013) “Dam and Levee Breaching Formulations in U.S. Army Corps of Engineers Finite Element Method Code Adaptive Hydraulics – 2-Dimensional Shallow Water Module” Dynamic Solutions, LLC. Knoxville, TN.

Savant, Gaurav (2015) “Three-Dimensional Shallow Water Adaptive Hydraulics (ADH-SW3): Turbulence Closure”. USACE ERDC/CHL CR-15-1, June 2015. Vicksburg, MS.

Savant, Gaurav and Berger, Charlie (2009) “Considerations for Modeling Flow Control Structures in Adaptive Hydraulics (ADH)” System Wide Water Resources Program Technical Note, ERDC-TN-SWWRP-09-3, 2009.

Savant,G.,G. L. Brown, and C. Berger, (2009) ”Friction in ice covered flows”. Proc. 33rd IAHR Congress: Water Engineering for a Sustainable Environment, Vancouver, Canada, International Association of Hydraulic Engineering & Research, 1300–1307.

Savant, Gaurav and Berger, Charlie (2010) “Intelligent Adaptive Time Step Control for Modeling Rapidly-Evolving Hydrodynamic Flows in Adaptive Hydraulics (ADH)”, System Wide Water Resources Program Technical Note, 2010.

Savant, Gaurav, R. Charlie Berger, Tate O. McAlpin, and Corey J. Trahan (2014). “Three-Dimensional Shallow Water Adaptive Hydraulics (ADH-SW3): Hydrodynamic Verification and Validation”. USACE ERDC TR-14-7. Vicksburg, MS.

Schureman, Paul “Manual of Harmonic Analysis and Prediction of Tides”, U.S. Department of Commerce, Coast and Geodetic Survey, Special Publication No. 98 Revised (1940) Edition, 1941.

Sharp, Jeremy A., Charles D. Lilttle, Gary L. Brown, Thad C. Pratt, Ronald E. Heath, Lisa C. Hubbard, Fred Pinkard, S. Keith Martin, Nathan Dwayne Clifton, David W. Perkey, and Naveen B. Ganesh (2013). “West Bay Sediment Diversion Effects”. USACE ERDC Technical Report, ERDC/CHL TR-13-15.

*Singh, V.P., and P.D. Scarlatos, 1988,”Analysis of Gradual Earth–Dam Failure,” Journal of Hydraulic Engineering, vol.114, no 1, p21-42

Smagorinsky, J. 1963. General circulation experiments with the primitive equations I. The basic experiment. *Monthly Weather Review* 91(3): 99–164.

Stansby, Peter K. (2004) “A Mixing Length Model for Shallow Turbulent Wakes” Shallow Flows, Jirka and Uijttewaal, eds. Taylor and Francis Group, London.

Swamee, P. (1992) "Sluice-Gate Discharge Equations". *Journal of Irrigation and Drainage Engineering*. ASCE. Vol. 118, Issue 1. January 1992.

Tate, J.N., McKinney, J.P, Pratt, T.C. (2009). "Salinas de San Pedro wetland Restoration Project". USACE ERDC Technical Report, TR-09-15.

Tate, J.N. and Ross, C.G. (2009). "Houston-Galveston Navigation Channels, Texas Project: Navigation Channel Sedimentation Study". USACE ERDC Technical report, TR-09-6.

Tate, J.N., Savant, G. and McVan D.C. (2012). "Rapid Response Numerical modeling of the 2010 Pakistan Flooding." Journal of Leadership and Management in Engineering, Vol. 12, No: 4.

Teeter, Allen Michael (2002) "Sediment Transport in Wind-Exposed Shallow, Vegetated Aquatic Systems" A Dissertation. Louisiana State University.

Von Thun, J.Lawrence, and David R.Gillette, 1990, Guidance on Breach Parameters, unpublished internal document, U.S.Bureau of Reclamation, Denver, Colorado, March 13,1990,17 p

Walton R., and Christensen, B. A. (1980). "Friction factors in storm surges over inland areas" *J. Waterw. Port, Coastal Ocean Div., Am. Soc. Civ. Eng.*, 106(2), 261-271.

Webel, G. and Schatzmann, M. (1984). "Transverse Mixing in Open Channel Flow" *Journal of Hydraulic Engineering*, Vol 110, No. 4, April 1984, pp 423-435.

Wu, Jin (1982). "Wind-Stress Coefficients Over Sea Surface From Breeze to Hurricane" *Journal of Geophysical Research,* Vol 87, No. C12, November 1982, pp 9704-9706.

Zienkiewicz, O.C and Zhu, J.Z "The Superconvergent Patch Recovery and *a Posteriori* Error Estimates. Part 1: The Recovery Technique", International Journal for Numerical Methods in Engineering, 33(7), 1331-1364, 2005.

Appendix: AdH Boundary Condition Cards (alphabetical)

BR BRC

DIRICHLET – BREACH BUREAU OF RECLAMATION

Field	Type	Value	Description
1	char	BR	Card type
2	char	BRC	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≥ 0	Breach failure time (seconds)
9	int	≥ 1	Side slope node furthest from breach (if side slope section)
10	int	≥ 1	Side slope node closest to breach (if side slope section)

BR FER

DIRICHLET – BREACH FEDERAL ENERGY REGULATORY COMMISSION

Field	Type	Value	Description
1	char	BR	Card type
2	char	FER	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	int	0 or 1	Engineer character – 0=non-engineered, 1=engineered/well compacted
9	real	≥ 0	Breach failure time (seconds)
10	int	≥ 1	Side slope node furthest from breach (if side slope section)
11	int	≥ 1	Side slope node closest to breach (if side slope section)

BR FRO

DIRICHLET – BREACH FROELICH

Field	Type	Value	Description
1	char	BR	Card type
2	char	FRO	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0.0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≥ 0.0	Breach failure time (seconds)
9	real	0.1≥#≤4.0	Exponent for main breach, controls the rate of the main breach progression
10	int	≥ 1	Side slope node furthest from breach (if side slope section)
11	int	≥ 1	Side slope node closest to breach (if side slope section)

BR JAI

DIRICHLET – BREACH JOHNSON AND ILLES

Field	Type	Value	Description
1	char	BR	Card type
2	char	JAI	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≥ 0	Breach failure time (seconds)
9	int	≥ 1	Side slope node furthest from breach (if side slope section)
10	int	≥ 1	Side slope node closest to breach (if side slope section)

BR MLM

DIRICHLET – BREACH MACDONALD AND LANDGRIDGE-MONOPOLIS

Field	Type	Value	Description
1	char	BR	Card type
2	char	MLM	Parameter
5	real	≥ 0.0	Maximum water depth above breach bottom (meters)
6	real	≥ 0.0	Reservoir volume (cubic meters)
7	real	#	Minimum breach elevation (meters)
8	real	#	Dam/Levee crest elevation (meters)

BR SAS

DIRICHLET – BREACH SINGH AND SNORRASON

Field	Type	Value	Description
1	char	BR	Card type
2	char	SAS	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0.0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	real	≤ 3600	Breach failure time (seconds - complete failure by 3600s)
9	int	≥ 1	Side slope node furthest from breach (if side slope section)
10	int	≥ 1	Side slope node closest to breach (if side slope section)

BR USR

DIRICHLET – USER DEFINED BREACH DISPLACEMENT

Field	Type	Value	Description
1	char	BR	Card type
2	char	USR	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the elevation of the string

BR VTG

DIRICHLET – BREACH VON THUN AND GILLETTE

Field	Type	Value	Description
1	char	BR	Card type
2	char	VTG	Parameter
3	int	≥ 1	String ID number (node)
4	int	0 or 1	Breach section – 0=main breach, 1=breach side slope
5	real	≥ 0	Width of main breach (meters)
6	real	#	Minimum breach elevation (meters)
7	real	#	Dam/levee crest elevation (meters)
8	int	0 or 1	Erosion character – 0=easily erodible, 1=erosion resistant
9	real	≥ 0	Breach failure time (seconds)
10	int	≥ 1	Side slope node furthest from breach (if side slope section)
11	int	≥ 1	Side slope node closest to breach (if side slope section)

CN CON

ANY GENERIC CONSERVATIVE CONSTITUENT

Field	Type	Value	Description
1	char	CN	Card type
2	char	CON	Parameter
3	int	≥ 1	Constituent ID number
4	real	> 0	Characteristic concentration

CN SAL

SALINITY (BAROCLINIC TRANSPORT)

Field	Type	Value	Description
1	char	CN	Card type
2	char	SAL	Parameter
3	int	≥ 1	The constituent ID number
4	real	> 0	Reference concentration (ppt)

CN TMP

TEMPERATURE (BAROCLINIC TRANSPORT)

Field	Type	Value	Description
1	char	CN	Card type
2	char	TMP	Parameter
3	int	≥ 1	The constituent ID number
4	real	> 0	Reference concentration (Celsius)
5	int	1, 2	Heat transfer though air/water interface (0 = no, 1 = yes)

CN VOR

VORTICITY TRANSPORT - BENDWAY CORRECTION

Field	Type	Value	Description
1	char	CN	Card type
2	char	VOR	Parameter
3	int	≥ 1	The constituent ID number
4	real	> 0	Normalization factor
5	real	≥ 0	A_s term, default is 0.0 which sets A_s = 5.0
6	real	≥ 0	D_s term, default is 0.0 which sets D_s = 0.5

DB LDE

DIRICHLET - STATIONARY LID ELEVATION

Field	Type	Value	Description
1	char	DB	Card type
2	char	LDE	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the elevation to be implemented

DB LDH

DIRICHLET - DEPTH OF WATER UNDER STATIONARY LID

Field	Type	Value	Description
1	char	DB	Card type
2	char	LDH	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the depth

DB LID

DIRICHLET - FLOATING STATIONARY OBJECT

Field	Type	Value	Description
1	char	DB	Card type
2	char	LID	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number that contains the draft of the lid

DB OVH

DIRICHLET - VELOCITY AND DEPTH

Field	Type	Value	Description
1	char	DB	Card type
2	char	OVH	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number for x-velocity component
5	int	≥ 1	Series ID number for y-velocity component
6	int	≥ 1	Series ID number for the depth

DB OVL

DIRICHLET – VELOCITY (2D)

Field	Type	Value	Description
1	char	DB	Card type
2	char	OVL	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Series ID number for x-velocity component
5	int	≥ 1	Series ID number for y-velocity component

DB RAD

DIRICHLET – SHORT WAVE RADIATION AND DEW POINT TEMPERATURE

Field	Type	Value	Description
1	char	DB	Card type
2	char	RAD	Parameter
3	int	≥ 1	Series ID number with short wave radiation in w/m^2
4	int	≥ 1	Series ID number that dew point temperature in Celsius

DB TRN

DIRICHLET BOUNDARY CONDITION - TRANSPORT

Field	Type	Value	Description
1	char	NB	Card type
2	char	TRN	Parameter
3	int	≥ 1	String ID number (node)
4	int	≥ 1	Constituent ID number
5	int	≥ 1	Series ID number that contains the constituent concentration (units dependent of the transport type)

EGS

EDGE STRINGS

Field	Type	Value	Description
1	char	EGS	Card type
2	int	≥ 1	ID number of the first node of an edge element
3	int	≥ 1	ID number of the second node of an edge element
4	int	≥ 1	String ID number

END

STOPPING THE MODEL

Field	Type	Value	Description
1	char	END	Close the model

FGT

FLAP GATE PARAMETERS

Field	Type	Value	Description
1	char	FGT	Card type
2	int	≥ 1	Flap gate number
3	int	= 1	1 – User specified parameters
		= 2	2 – Automatic computation (not yet implemented)
4	int	≥ 1	String upstream of flap (node)
5	int	≥ 1	String downstream of flap (node)
6	int	≥ 1	Flap string on the upstream (edge)
7	int	≥ 1	Flap string on the downstream (edge)
8	real	≥ 0	Coefficient A
9	real	≥ 0	Coefficient B
10	real	≥ 0	Coefficient C
11	real	≥0	Coefficient D
12	real	≥0	Coefficient E
13	real	≥0	Coefficient F
14	real	≥0	Length of flap gate

FLP

NUMBER OF FLAP GATES

Field	Type	Value	Description
1	char	FLP	Card type
2	int	≥ 1	Number of flap gates

FLX

FLOW OUTPUT

Field	Type	Value	Description
1	char	FLX	Parameter
2	int	> 0	String ID number for the mid string or edge string for which flow is to be output

FR BRD

1D INTERNAL FRICITON: BRIDGE DECK

Field	Type	Value	Description
1	char	FR	Card type
2	char	ICE	Parameter
3	int	> 0	Midside string ID number
4	real	> 0.0	Elevation of the bridge deck (h_B is calculated using the difference of this value and the local bed elevation)
5	real	> 0.0	Thickness of the bridge deck (t_B)

FR BRH

ICE BED ROUGHNESS HEIGHT

Field	Type	Value	Description
1	char	FR	Card type
2	char	BRH	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Bed roughness height (k_{BED})

FR EDO

UN-SUBMERGED RIGID VEGETATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	URV	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Bed Roughness Height (not including the obstructions) (k_B)
5	real	> 0.0	Canopy Roughness Height (k_C)
6	real	> 0.0	Average diameter of the obstructions (d)
7	real	> 0.0	Average height of the obstructions (h_{OBS})
8	real	> 0.0	Average density of the obstructions (number/unit area) (m)

FR ERH

LOG PROFILE ROUGHNESS: EQUIVALENT ROUGHNESS HEIGHT

Field	Type	Value	Description
1	char	FR	Card type
2	char	ERH	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	Roughness height (k)

FR ICE

ICE THICKNESS

Field	Type	Value	Description
1	char	FR	Card type
2	char	ICE	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Ice thickness
5	real	> 0.0	Ice density
6	int	= 0	0 – stationary
		= 1	1 – moving ice (not implemented yet)

FR IRH

ICE ROUGHNESS

Field	Type	Value	Description
1	char	FR	Card type
2	char	IRH	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Ice roughness height (k_{ICE})

FR MNC

CLASSIC MANNING'S EQUATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	MNC	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	Manning's n

FR MNG

LOG PROFILE ROUGHNESS: MANNING'S N

Field	Type	Value	Description
1	char	FR	Card type
2	char	MNG	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	Manning's n

FR SAV

SUBMERGED AQUATIC VEGETATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	SAV	Parameter
3	int	> 0	String ID number
4	real	≥ 0.0	The roughness height of the SAV canopy (k)
5	real	≥ 0.0	Undeflected stem height (hs_{sav})

FR SDK

1D INTERNAL FRICITON: SUBMERGED DIKE

Field	Type	Value	Description
1	char	FR	Card type
2	char	ICE	Parameter
3	int	> 0	Midside string ID number
4	real	> 0.0	Height of the dike (above the bed) (a)

FR URV

UN-SUBMERGED RIGID VEGETATION

Field	Type	Value	Description
1	char	FR	Card type
2	char	URV	Parameter
3	int	> 0	String ID number
4	real	> 0.0	Bed Roughness Height (not including the stems) (k)
5	real	> 0.0	Average stem diameter (d)
6	real	> 0.0	Average stem density (m)

IP ITL

INCREMENT TOLERANCE

Field	Type	Value	Description
1	char	IP	Card type
2	char	ITL	Parameter
3	real	≥ 0	Tolerance for maximum allowable change in the velocity and depth solutions between non-linear iterations

IP MIT

LINEAR ITERATIONS

Field	Type	Value	Description
1	char	IP	Card type
2	char	MIT	Parameter
3	int	≥ 1	Maximum number of linear iterations per non-linear iteration by the interative solver. If the internal linear tolerance (0.0001* NTL) is not met in the maximum linear iterations, the solution stops and the timestep size is reduced.

IP NIT

NON-LINEAR ITERATIONS

Field	Type	Value	Description
1	char	IP	Card type
2	char	NIT	Parameter
3	int	≥ 1	Number of non-linear iterations per timestep, if at NIT the tolerance is not satisfied AdH will reduce the timestep and recalculate

IP NTL

NON-LINEAR TOLERANCE

Field	Type	Value	Description
1	char	IP	Card type
2	char	NTL	Parameter
3	real	≥ 0	Tolerance for maximum allowable solution residual for the non-linear iterations (perfect convergence has a residual of 0)

MDS

MID STRINGS

Field	Type	Value	Description
1	char	MDS	Card type
2	int	≥ 1	ID number of the first node of an edge element
3	int	≥ 1	ID number of the second node of an edge element
4	int	≥ 1	String ID number

MP COR

CORIOLIS LATITUDE

Field	Type	Value	Description
1	char	MP	Card type
2	char	COR	Parameter
3	int	≥ 1	Material type ID number
4	real	-90 ≤ # ≤ 90	Latitude

MP DF

TURBULENT DIFFUSION RATE

Field	Type	Value	Description
1	char	MP	Card type
2	char	DF	Parameter
3	int	≥ 1	Material type ID number
4	int	≥ 0	Constituent ID number
5	real	≥ 0.0	Turbulent diffusion rate

MP DTL

WETTING/DRYING LIMIT (2D)

Field	Type	Value	Description
1	char	MP	Card type
2	char	DTL	Parameter
3	real	≥ 0.0	wetting and drying “depth” (stabilization term), default is 0.0

MP EEV

ESTIMATED EDDY VISCOSITY

Field	Type	Value	Description
1	char	MP	Card type
2	char	EEV	Parameter
3	int	≥ 1	Material type ID number
4	real	K	Value is dependent on the method (see above)
5	int	1, 2, 3, or 4	1 for isotropic (legacy) formulation 2 for anisotropic formulation 3 for Smagorinsky 4 for Stansby

MP EVS

CONSTANT EDDY VISCOSITY (2D)

Field	Type	Value	Description
1	char	MP	Card type
2	char	EVS	Parameter
3	int	≥ 1	Material type ID number
4	real	> 0	E_{xx}
5	real	> 0	E_{yy}
6	real	> 0	E_{xy}

MP G

GRAVITATIONAL ACCELERATION

Field	Type	Value	Description
1	char	MP	Card type
2	char	G	Parameter
3	real	≥ 0	Value of gravity induced acceleration (L/T^2)

MP ML

REFINEMENT LEVELS

Field	Type	Value	Description
1	char	MP	Card type
2	char	ML	Parameter
3	int	≥ 1	Material type ID number
4	int	≥ 0	Maximum number of refinement levels

MP MU

KINEMATIC MOLECULAR VISCOSITY

Field	Type	Value	Description
1	char	MP	Card type
2	char	MU	Parameter
3	real	≥ 0	Uniform background viscosity (kinematic molecular viscosity, units L^2/T)

MP MUC

MANNING'S UNITS CONSTANT

Field	Type	Value	Description
1	char	MP	Card type
2	char	MUC	Parameter
3	real	> 0.0	Coefficient (1.486 for English units, 1.0 for SI standard)

MP NVM

NO VORTICITY TRANSPORT BY MATERIAL

Field	Type	Value	Description
1	char	MP	Card type
2	char	NVM	Parameter
3	int	≥ 1	The material ID number

MP RHO

DENSITY

Field	Type	Value	Description
1	char	MP	Card type
2	char	RHO	Parameter
3	real	≥ 0	Density (M/L^3)

MP SRT

FLOW REFINEMENT TOLERANCES

Field	Type	Value	Description
1	char	MP	Card type
2	char	SRT	Parameter
3	int	≥ 1	Material type ID number
4	real	≥ 0	Error tolerance for the refinement terms

MP TRT

TRANSPORT CONSTITUENT REFINEMENT TOLERANCE

Field	Type	Value	Description
1	char	MP	Card type.
2	char	TRT	Parameter.
3	int	≥ 1	Material type ID number
4	int	≥ 1	Constituent ID number
5	real	≥ 0	Error tolerance for refinement terms

MP WND ATT

WIND ATTENUATION

Field	Type	Value	Description
1	char	MP	Card type
2	char	WND	Parameter
3	char	ATT	Parameter
4	int	≥ 1	Material type ID number
5	real	0 ≤ #	Fraction applied to wind stress (default is 1.0)

MP WND STR

WIND STRESS

Field	Type	Value	Description
1	char	MP	Card type
2	char	WND	Parameter
3	char	STR	Parameter
4	int	≥ 1	Material type ID number
5	real	0,1,2	Wind transform (0 = no transform 1 = Wu 2 = Teeter)

MTS

MATERIAL STRINGS

Field	Type	Value	Description
1	char	MTS	Card type

2	int	≥ 1	Material type ID number
3	int	≥ 1	String ID number

NB DIS

NATURAL BOUNDARY CONDITION - TOTAL DISCHARGE

Field	Type	Value	Description
1	char	NB	Card type
2	char	DIS	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Series ID number containing the total discharge across the string; positive in

NB OTW

NATURAL BOUNDARY CONDITION - WATER SURFACE ELEVATION

Field	Type	Value	Description
1	char	NB	Card type
2	char	OTW	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Series ID number that contains the time series of the water surface elevation

NB OUT

FLOW OUTPUT FROM INSIDE THE GRID

Field	Type	Value	Description
1	char	NB	Card type
2	char	OUT	Card type
3	int	≥ 1	Outflow edge string
4	int	≥ 1	Inflow edge string
5	int	≥ 1	Series ID number of the outflow

NB OVL

ATMOSPHERIC SOURCE

Field	Type	Value	Description
1	char	NB	Card type
2	char	SOURCE	Parameter
3	int	≥ 1	String ID number (material or face)

4	int	≥ 1	Series ID number that contains the time series of the source values (unit flow – L/T)

NB OVL

NATURAL BOUNDARY CONDITION - FLOW

Field	Type	Value	Description
1	char	NB	Card type
2	char	VEL	Parameter
3	int	≥ 1	String ID number (2D edge or 2D material)
4	int	≥ 1	Series ID number containing the flow data; for material strings the series values represent the flow per unit area (L/T - positive in); for edge strings the series values represent the flow per unit width (L^2/T - positive in)

NB SDR

STAGE DISCHARGE BOUNDARY

Field	Type	Value	Description
1	char	NB	Card type
2	char	SDR	Parameter
3	int	≥ 1	String ID number
4	real	≥ 0	Coefficient A
5	real	≥ 0	Coefficient B
6	real	≥ 0	Coefficient C
7	real	≥ 0	Coefficient D
8	real	≥ 0	Coefficient E

NB SPL

NATURAL BOUNDARY CONDITION - SPILLWAY

Field	Type	Value	Description
1	char	NB	Card type
2	char	SPL	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Series ID number that contains the time series of the percent (%) flow out.

NB TID

TIDAL CONSTITUENT BOUNDARY CONDITION

Field	Type	Value	Description
1	char	NB	Card type
2	char	TID	Parameter
3	int	≥ 1	String ID number (edge)

NB TRN

NATURAL BOUNDARY CONDITION - TRANSPORT

Field	Type	Value	Description
1	char	NB	Card type
2	char	TRN	Parameter
3	int	≥ 1	String ID number (edge)
4	int	≥ 1	Constituent ID number
5	int	≥ 1	Series ID number that contains the constituent concentration (units dependent of the transport type)

NDS

NODE STRINGS

Field	Type	Value	Description
1	char	NDS	Card type
2	int	≥ 1	ID number of a node with a Dirichlet condition
3	int	≥ 1	String ID number

OB OF

NATURAL OUTFLOW

Field	Type	Value	Description
1	char	OB	Card type
2	char	OF	Parameter
3	int	≥ 1	String ID number (edge)

OC

OUTPUT

Field	Type	Value	Description
1	char	OC	Parameter
2	int	> 0	Series ID number that contains the time steps to be output

OFF

DEACTIVATE STRING

Field	Type	Value	Description
1	char	OFF	Card type
2	int	> 0	String ID number

OP BLK

BLOCK SPECIFICATION FOR PRE-CONDITIONER

Field	Type	Value	Description
1	char	OP	Card type
2	char	BLK	Parameter
3	int	> 0	Number of blocks per processor, used to perform pre-conditioning

OP BT

VESSEL MOVEMENT LIBRARY INCLUSION (ENABLE VESSEL MOVEMENT)

Field	Type	Value	Description
1	char	OP	Card type
2	char	BT	Parameter

OP BTS

ENABLE VESSEL ENTRAINMENT

Field	Type	Value	Description
1	char	OP	Card type
2	char	BTS	Parameter

OP DAM

DAM BREAK STABILIZATION

Field	Type	Value	Description
1	char	OP	Card type
2	char	DAM	Parameter

OP INC

MEMORY INCREMENT

Field	Type	Value	Description
1	char	OP	Card type
2	char	INC	Parameter
3	int	> 0	Incremental memory allocation

OP NF2

SW2 GRADIENTS

Field	Type	Value	Description
1	char	OP	Card type
2	char	NF2	Parameter

OP PRE

PRE-CONDITIONER SELECTION

Field	Type	Value	Description
1	char	OP	Card type
2	char	PRE	Parameter
3	int	3 ≥ # ≥ 0	Preconditioner value 0 No pre-conditioning **1 one level Additive Schwarz pre-conditioning** 2 two level Additive Schwarz pre-conditioning 3 two level Hybrid pre-conditioning

OP SW2

2D SHALLOW WATER PROBLEMS

Field	Type	Value	Description
1	char	OP	Card type
2	char	SW2	Specifies 2-D shallow water problem

OP TEM

SECOND ORDER TEMPORAL TERM

Field	Type	Value	Description
1	char	OP	Card type
2	char	TEM	Parameter
3	real	1 ≥ # ≥ 0	Coefficient for the second order temporal scheme

OP TPG

PETROV-GALERKIN COEFFICIENT

Field	Type	Value	Description
1	char	OP	Card type
2	char	TPG	Parameter
3	real	0.5 ≥ # ≥ 0	Coefficient for the Petrov-Galerkin equation

OP TRN

TRANSPORT EQUATIONS

Field	Type	Value	Description
1	char	OP	Card type
2	char	TRN	Parameter
3	int	≥ 0	Total number of transported materials

OP WAV

SHORT WAVE STRESSING

Field	Type	Value	Description
1	char	OP	Card type
2	char	WAV	Parameter

OS

AUTO-BUILD OUTPUT SERIES

Field	Type	Value	Description
1	char	OS	Card type
2	int	> 0	ID number of the series
3	int	> 0	Number of points in the series
4	int	> 0	Output units (Units 0 = seconds, 1 = minutes, 2 = hours, 3 = days, 4 = weeks)

PC ADP

ADAPTED MESH PRINTING

Field	Type	Value	Description
1	char	PC	Card type
2	char	ADP	Adaptive mesh printing turned on - omit to turn off

PC ELM

NUMERICAL FISH SURROGATE OUTPUT

Field	Type	Value	Description
1	char	PC	Card type
2	char	ELM	Numerical Fish Surrogate output (optional card)

PC LVL

SCREEN OUTPUT FORMAT

Field	Type	Value	Description
1	char	PC	Card type
2	char	LVL	Screen output format – level 0 is default

3	int	0, 1, 2	0 gives short column format; 1 gives long column format; 2 gives original AdH format (non-tabular)

PC MEO

MASS ERROR OUTPUT PRINT CONTROL

Field	Type	Value	Description
1	char	PC	Card type
2	char	MEO	MEO output from AdH simulation
3	int	0, 1	0 is the default (does not print), 1 prints the mass error output to screen

SLS

SLUICE GATE PARAMETERS

Field	Type	Value	Description
1	char	SLS	Card type
2	int	≥ 1	Sluice Gate Number
3	int	≥ 1	String upstream of sluice gate (node)
4	int	≥ 1	String downstream of sluice gate (node)
5	int	≥ 1	Sluice string on upstream (edge)
6	int	≥ 1	Sluice string on downstream (edge)
7	real	≥ 0	Length of sluice gate (*b* in the previous equations)
8	int	≥ 1	Time series defining the sluice gate opening over time (*a* in the previous equations)

SLUICE

NUMBER OF SLUICE GATES

Field	Type	Value	Description
1	char	SLUICE	Card type
2	int	≥ 1	Number of sluices

TC ATF

AUTOMATIC TIMESTEP FIND

Field	Type	Value	Description
1	char	TC	Card type
2	char	ATF	Parameter
3	int	> 0	Minimum time step size.

4	int	> 0	Maximum time step size series

TC IDT

TIME STEP SIZE

Field	Type	Value	Description
1	char	TC	Card type
2	char	IDT	Parameter
3	int	> 0	Series ID number containing the length of timestep (Δt).

TC STD

STEADY STATE

Field	Type	Value	Description
1	char	TC	Card type
2	char	STD	Parameter
3	int	> 0	Minimum time step size.
4	int	> 0	Maximum time step size

TC T0

START TIME

Field	Type	Value	Description
1	char	TC	Card type
2	char	T0	Parameter
3	real	> 0	Start time of the model
4	int	#	Units (optional; 0 = seconds, 1 = minutes, 2 = hours, 3 = days, 4 = weeks)

TC TF

FINAL TIME

Field	Type	Value	Description
1	char	TC	Card type
2	char	TF	Parameter
3	real	> 0	End time of the model
4	int	#	Units (optional; 0 = seconds, 1 = minutes, 2 = hours, 3 = days, 4 = weeks)

WER

NUMBER OF WEIRS

Field	Type	Value	Description
1	char	WER	Card type
2	int	≥ 1	Number of weirs

WRS

WEIR PARAMETERS

Field	Type	Value	Description
1	char	WRS	Card type
2	int	≥ 1	Weir Number
3	int	≥ 1	String upstream of weir (node)
4	int	≥ 1	String downstream of weir (node)
5	int	≥ 1	Weir string on upstream (edge)
6	int	≥ 1	Weir string on downstream (edge)
7	real	≥ 0	Length of weir
8	real	≥ 0	Weir crest elevation
9	real	≥ 0	Weir height

XY1

X-Y SERIES

Field	Type	Value	Description
1	char	SERIES	Card type
2	char	BC	Series type
3	int	> 0	ID number of the series
4	int	> 0	Number of points in the series
5	int	> 0	Input units. (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)
5	int	> 0	Output units (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)

XY2

X-Y-Y SERIES

Field	Type	Value	Description
1	char	XY2	Card type
2	int	> 0	ID number of the series
3	int	> 0	Number of points in the series

4	int	> 0	Input units. (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)
5	int	> 0	Output units (0 = seconds; 1 = minutes; 2 = hours; 3 = days; and 4 = weeks)

Currently, only the data that is to be used for wind series is to be input via the X-Y-Y series.

XYC

WIND STATION COORDINATES

Field	Type	Value	Description
1	char	XYC	Card type
2	int	≥ 1	ID number of the series to which it is associated
3	real	#	X coordinate of the wind station
4	real	#	Y coordinate of the wind station